Discriminative Articulatory Feature-based Pronunciation Models with Application to Spoken Term Detection

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Rohit Prabhavalkar, B.E., M.S.

Graduate Program in Computer Science & Engineering

The Ohio State University

2013

Dissertation Committee:

Professor Eric Fosler-Lussier, Advisor

Professor Brian Kulis

Professor Karen Livescu

Professor William Schuler

ABSTRACT

Conversational speech is characterized by large amounts of variability; variation in accents, pronunciation and disfluencies continue to present challenges for speech recognition systems. Speech recognition systems must account for this variability if they are to be successfully deployed in real-world environments. Traditional speech recognition approaches, based on the so-called *'beads on a string'* approach have a number of drawbacks when it comes to modeling the variability in conversational speech.

In recent work, articulatory feature-based pronunciation models have been proposed as alternatives to phone-based representations, and have been shown to improve performance in various studies. These models are grounded in linguistic theories and attempt to explain the variation observed in conversational speech by hypothesizing it to be produced in part as a result of the relative asynchrony between the speech articulators.

The main contributions of this thesis are the development of discriminative articulatory feature-based pronunciation models, and the application of these models to the task of detecting words or phrases in conversational speech. We first develop factored conditional random field models of the articulatory feature streams, which explicitly account for the ability of the speech articulators to desynchronize in conversational speech. Additionally, we describe how exact inference can be performed efficiently in the proposed models by exploiting deterministic task-specific constraints. In experimental evaluations, we find

that the proposed *discriminative* conditional random field models outperform previously proposed *generative* dynamic Bayesian network models for the task.

We then apply the proposed articulatory feature-based pronunciation models to the problem of spoken term detection: detecting whether and where specific words or phrases are uttered in conversational speech. We conduct detailed evaluations to determine the effectiveness of the proposed techniques in low-resource settings where transcribed training data are limited and find that the proposed articulatory feature-based models improve performance over phone-based models in a number of settings. Additionally, in many instances, the information contained in the articulatory feature-based pronunciation models appears to be complementary to the phone-based pronunciation models allowing us to improve performance through model combination. Finally, we end the thesis by describing how the proposed spoken term detection approach can be adapted to leverage existing spoken term detection systems based on large vocabulary continuous speech recognizers, if available, in order to improve system running time and performance.

For my family.

ACKNOWLEDGMENTS

This thesis would not have been possible without the help and support of a number of people. First and foremost, I would like to thank my advisor, Eric Fosler-Lussier, for his constant support and encouragement. Eric is the perfect advisor I could have asked for. Over the past few years, has has given me the freedom to explore research areas based on my interests while always being available to provide guidance when I found myself struggling to make progress on a problem. Often I would enter our weekly meeting unsure of how to proceed, only to leave the meeting with a new sense of clarity and my mind brimming with ideas. Above all, I would like to thank him for always pushing me to be the best researcher that I could be. Eric has had a huge role in shaping my abilities as a researcher and I am forever grateful to have had the opportunity of working with him.

I owe a debt of gratitude to Karen Livescu for her advice and suggestions throughout my graduate studies. I thank her for setting up my visit to TTI-Chicago for the summer in 2010, and for sharing her detailed knowledge on articulatory feature-based pronunciation models. Karen has an amazing ability to succinctly and beautifully express ideas through the written word and her clarity and attention to detail continue to push me to hone my own writing skills. I am immensely thankful to Karen for all of her inputs and suggestions on all of my research.

I would like to thank Joseph (Yossi) Keshet for his advice on all things related to machine learning and for setting me down the path of discriminative spoken term detection.

I would like to thank Yossi for the many spirited discussions that we had on a various research topics. Yossi has the uncanny ability to take seemingly complex ideas and describe them in an extremely intuitive manner. A large part of the work on spoken term detection in this thesis would not have been possible without his guidance and suggestions, and I am extremely thankful to him for this.

I would also like thank John Josephson (JJ) for allowing me to work with him during my first year at OSU and for our wide-ranging and wonderful discussions on a variety of topics. JJ helped make me feel at home when I first started as a graduate student at OSU and I shall always be grateful to him for that.

I would like to thank the members of the Slate Lab – Billy Hartmann, Yanzhang (Ryan) He, Illana Heintz, Preethi Jyothi, Joo-Kyung Kim, Josh King, Yi Ma, Jeremy Morris, Preethi Raghavan, Darla Shockley, and Tim Weale – for all of the wonderful years that I spent as part of the lab. Getting to know all of you better has been a great pleasure and I will sorely miss our lab-lunches. I thank Jeremy for his tremendous help with getting up to speed on various aspects of automatic speech recognition and discriminative machine learning techniques when I first joined the lab. I thank Billy for his help on various aspects of speech recognition, but above all for his unique sense of humor. I particularly want to thank Ryan for his help on the Cantonese spoken term detection experiments and for helping improve my understanding of various research topics through our discussions. Most of all, I would like to thank Preethi Jyothi for patiently listening to my research ideas over the years, and for suggesting improvements and pointing out flaws where appropriate.

I would like to thank Michael Mandel and members of the Computational Linguistics (Clippers) reading group – Steve Boxwell, Chris Brew, Marie-Catherine De Marneffe, Jon Dehdari, Micha Elsner, Dominic Espinoza, D. J. Hovermale, Dave Howcroft, Evan Jaffe,

Dennis Mehay, Rajakrishnan (Raja) Rajakumar, William Schuler, Marty Van Schijndel, and Mike White – for all of their helpful comments and suggestions during my research presentations. Thanks are also due to members of the perception and neurodynamics lab at OSU – Kun Han, Ke Hu, Zhaozhang Jin, Arun Narayanan, Yuxuan Wang, John Woodruff, and Xiaojia Zhao – for their comments on my research. In particular, I would like to thank Zhaozhang Jin, DeLiang (Leon) Wang and John Woodruff for our research collaborations.

I would like to thank Jasha Droppo at Microsoft Research and Tara Sainath, Bhuvana Ramabhadran, David Nahamoo and Dimitri Kanevsky at IBM research for allowing me the opportunity to work on interesting research problems during summer internships in 2011 and 2012 respectively.

Through visits to TTI-C and summer internships at Microsoft Research and IBM Research I had the opportunity to meet and interact with an amazing set of students. I thank Raman Arora, Jackie Cheung, Keith Godin, Brian Hutchinson, Gabrielle Knight, Andrew Maas, Abdel-rahman Mohamed, Amr Mousa, Arild Næss, Suman Ravuri, Hao Tang, and Ainur Yessenalina for making my time spent away from Columbus extremely enjoyable. I would particularly like to thank Arild for ensuring that my trips to Chicago were fun-filled and for our research collaborations.

I would like to thank the members of my candidacy exam committee – William Schuler, Mikhail Belkin and James Davis – for their tough questions and helpful comments, which served to improve the quality of the thesis. I would also like to thank William Schuler, Brian Kulis, Karen Livescu, and Cynthia Clopper for serving on my dissertation committee and for their feedback on earlier drafts of this thesis.

Thanks are also due to the members of the administrative staff in the department of Computer Science and Engineering – Catrena Collins, Tamera Cramer, Don Havard, Lynn

Lyons, Meg Murnane, Kitty Reeves, and Carrie Stein – for always being available to answer my many questions about department and graduate school policies and requirements.

I would like to thank the National Science Foundation (NSF) and the Intelligence Advanced Research Project Activity (IARPA) for supporting my graduate research through various grants; the research presented in the thesis was supported by the under NSF CAREER grant IIS-0643901, NSF grants IIS-0905420, and IIS-0905633, and by IARPA via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0014.

I thank all of my friends in Columbus for making it a home away from home for the past few years; I will always have fond memories of the time that I spent here. I thank Hari Ravindran, Tapopriya Majumdar, Pawas Ranjan, Chaitanya Shivade, and members of the Columbus Go club for making sure that my weekends in Columbus were full of fun.

Finally, I would like to thank my family for all of their sacrifices, and for their constant love and support, without which none of this would have been possible.

VITA

April 30, 1985	Born in Mumbai, Maharashtra, India.
August, 2007	B.E., Computer Engineering University of Pune, Pune, India.
June, 2012	M.S., Computer Science and Engineering The Ohio State University, OH, USA.

PUBLICATIONS

Journal Articles

E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional Random Fields in Speech, Audio, and Language Processing," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1054–1075, May 2013.

Conference Papers and Books

R. Prabhavalkar, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative Articulatory Models for Spoken Term Detection in Low-Resource Conversational Settings", *in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013.

R. Prabhavalkar, T. N. Sainath, D. Nahamoo, B. Ramabhadran, and D. Kanevsky, "An Evaluation of Posterior Modeling Techniques for Phonetic Recognition", *in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013.

R. Prabhavalkar, J. Keshet, K. Livescu, and E. Fosler-Lussier, "Discriminative Spoken Term Detection with Limited Data", *in Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, Portland, USA, 2012.

R. Prabhavalkar, and J. Droppo, "A Chunk-based Phonetic Score for Mobile Voice Search", *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012.

R. Prabhavalkar, E. Fosler-Lussier, and K. Livescu, "A Factored Conditional Random Field Model for Articulatory Feature Forced Transcription", *in IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Hawaii, USA, 2011.

A. Næss, K. Livescu, and R. Prabhavalkar, "Articulatory Feature Classification using Nearest Neighbors", *in Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, Florence, Italy, 2011.

J. Woodruff, R. Prabhavalkar, E. Fosler-Lussier, and D. L. Wang,"Combining Monaural and Binaural Evidence for Reverberant Speech Segregation", *in Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, Makuhari, Japan, 2010.

R. Prabhavalkar, P. Jyothi, W. Hartmann, J. Morris, and E. Fosler-Lussier, "Investigations into the Crandem Approach to Word Recognition", *in Proceedings of the annual conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Los Angeles, USA, 2010.

R. Prabhavalkar, and E. Fosler-Lussier, "Backpropagation Training for Multilayer Conditional Random Field based Phone Recognition", *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, USA, 2010.

R. Prabhavalkar, Z. Jin, and E. Fosler-Lussier, "Monaural Segregation of Voiced Speech using Discriminative Random Fields", *in Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, Brighton, UK, 2009.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in Artifical Intelligence: Professor Eric Fosler-Lussier

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1: INTRODUCTION

Over the past few decades, the state-of-the-art in speech recognition – the technology that allows spoken utterances to be automatically converted into written sentences using computational systems – has continued to improve. Spurred, in part, by improvements in the field of machine learning, our ability to build increasingly sophisticated systems that tackle challenging problems in automatic speech recognition (ASR) has enabled large-scale deployment of speech recognition engines in commercial devices and the market for such products is likely to grow dramatically over the coming years. The rapid progress that ASR technology has made in the past few decades can be readily observed in the data released by the National Institute of Standards and Technology (NIST), which has been conducting evaluations of ASR systems for a number of years. As can be seen in Figure 1.1, ASR systems are comparable to human performance on relatively simple tasks, such as the recognition of read speech; performance on more challenging tasks, recognizing conversational speech for example, remains significantly worse than human performance to the present day.

Given the statistics presented in Figure 1.1, a natural question presents itself: Why does this gap exist? What are the underlying causes that make machine recognition of speech such a hard problem? The task of speech recognition is challenging because of the large amount of variability that exists in human speech. Brief reflection on our everyday conversations reveals just how marvelous this ability really is. We routinely communicate in *noisy environments*, we have the ability to recognize *many accents and dialects that*

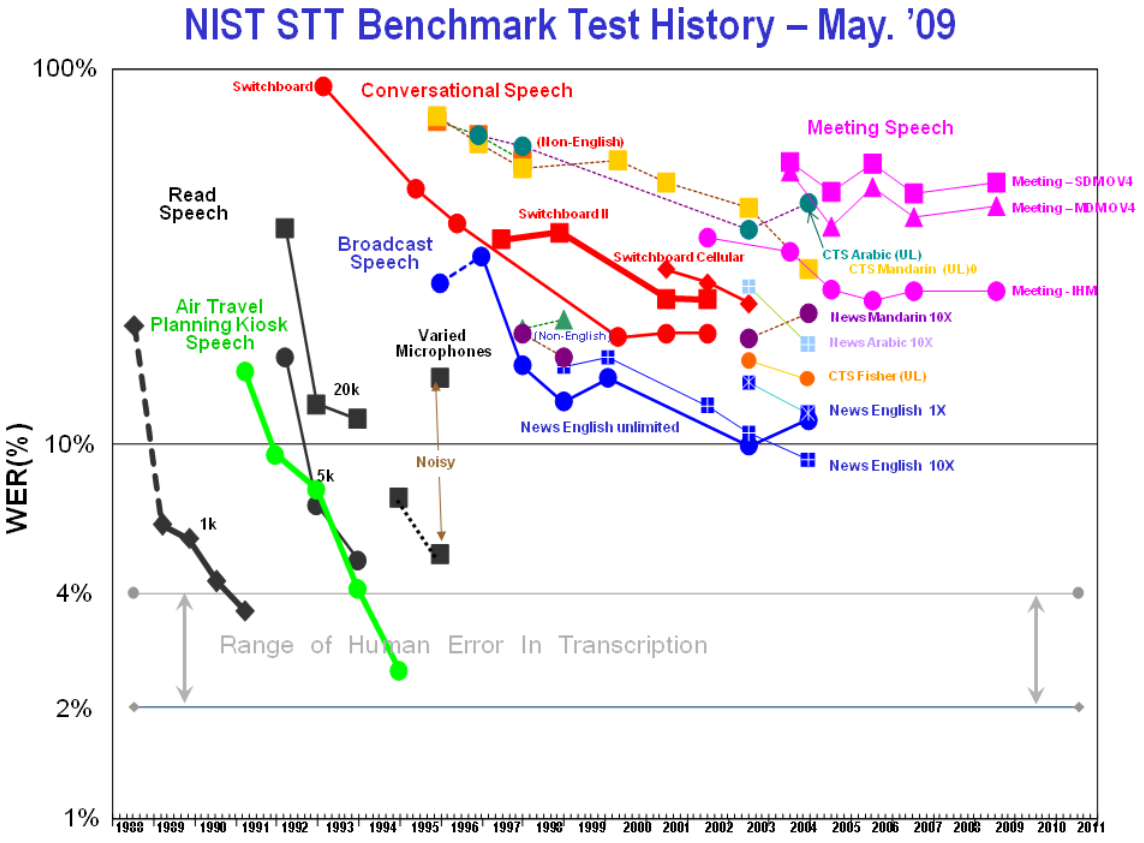

Figure 1.1: Results of NIST evaluations over the years reproduced from (http://www.itl.nist.gov/iad/mig/publications/ASRhistory/). Each point on the graph represents the performance of the best system in an ASR evaluation (Y-axis) conducted by NIST in that particular year (X-axis). Note that the scale on the Y-axis is logarithmic. The error rates of state-of-the-art systems in less challenging domains, e.g., read speech, is close to the level of human performance. However, performance in more challenging domains, such as the recognition of conversational speech, is significantly worse than human performance.

are not our own; when confronted with utterances that are *potentially confusable both acoustically as well as semantically*, we seem to have little or no trouble deriving the correct interpretation in most circumstances.[1] Bridging the gap between human performance and computational ASR technologies [Lippmann, 1997] is essential if these systems are to be widely deployed in everyday environments replete with background noise, variations in accent, pronunciation, speaking rate and other confounding factors.

The present thesis represents a step in the direction of making ASR systems more robust to variation by addressing the pronunciation variability encountered in conversational speech. Speech produced in conversational settings exhibits large amounts of variability in the pronunciation of words, making it particularly challenging for speech recognition systems [Farnetani and Recasens, 2012]. In conventional ASR systems, this variability is modeled only indirectly and thus these systems are limited in their ability to fully account for this variation. The thesis builds upon previous work in the area ([Browman and Goldstein, 1992; Deng et al., 1997; Livescu, 2005] inter alia) by developing models that directly account for the cause of pronunciation variation in conversational speech: the result of the interactions produced by the relative motion of the speech articulators. The models of pronunciation developed in the thesis are firmly grounded in linguistic theories of speech production and have shown promise over traditional approaches in prior work.

In this introductory chapter, we begin with a broad overview of the problem that forms the focus of the rest of the thesis: the pronunciation variability in conversational speech, and techniques for addressing this variability in ASR systems. Towards this end, in Section 1.1 we briefly discuss some of the challenges of recognizing conversational speech,

[1] My favorite example in this regard comes in the form of the following two phrases: "recognize speech" (/r eh k ah **g** n ay **z** s **p** iy ch/) and "wreck a nice beach" (/r eh k ah n ay s **b** iy ch/). The reader may rest assured that no beaches were damaged in the production of this thesis.

while briefly reviewing some of the techniques for addressing this variability that have been previously proposed in Section 1.2. In Section 1.3, we motivate pronunciation modeling at the sub-phonetic level; a detailed discussion is deferred until Chapter 2. We end this chapter with an overview of the organization of the rest of this thesis in Section 1.4.

1.1 Variability in Speech: Implications for ASR

Benzeghiba et al. [2007] broadly identify four sources of variability in speech. The identified sources are not necessarily disjoint, with some causes attributable to multiple classes, but nevertheless this taxonomy is a useful starting point for our discussions. The first source of variability identified by Benzeghiba et al. may be attributed to *physiological or behavioral differences* between speakers: the acoustic realization of the speech signal may vary based on the characteristics of the speakers vocal tract and other articulators, the speaker's environment or the speaker's mood. These factors directly influence the realized acoustic signal resulting in inter-speaker variation. Roughly speaking, this is what makes your speech *yours*, and differentiates it from everyone else's. The second source of variability can be attributed to an attempt by the speaker to convey *high-level information* such as emphasizing or questioning or to convey emotion. Such variability shows itself, for example, in intonational differences in speech. Another source of variability may be due to the consequence of *sociological factors* that may result in differences in the grammatical structure of the spoken language, for example based on the speakers level of knowledge of the language, non-nativeness etc. The fourth source of variability, the one that we shall be chiefly concerned with, is the variability caused by *pronunciation variation*: instances

where the pronunciation of the words (as observed in the *surface acoustic realization in terms of phonemes*[2]) is altered.

1.1.1 Challenges of Recognizing Conversational Speech

Words uttered in conversational speech are not produced in isolation; the effect of surrounding words impacts pronunciation [Farnetani and Recasens, 2012]. Take for instance the phrase "green pear". In conversational speech, the final nasal sound (/n/)[3] is followed by a bilabial stop (/p/; forming a complete closure of the vocal tract at the lips, followed by a *burst* of energy when the closure is released.) which might result in the phrase being acoustically realized as "gree*m* pear". Conversational speech contains more disfluencies than those observed in carefully articulated speech (e.g., speech produced during the reading of a document). Such disfluencies manifest themselves as false starts, repetitions and filled pauses. Another characteristic of conversational speech, which distinguishes it from carefully articulated speech, is that pronunciation of words is often 'sloppy' resulting in reduced articulation of phonemes. As a result, the observed pronunciations of words in conversational speech tend to differ markedly from the canonical pronunciation as may be expected according to a dictionary.

There is evidence that speaking style has an impact on the performance of speech recognition system; in particular, that spontaneous conversational speech is 'harder' for

[2]We differentiate the *canonical* or *target* pronunciation of a word, such as might be present in a dictionary, from the *surface* or *realized* pronunciation of the word, corresponding to the sequence of phoneme sounds a trained linguist might assign to a given segment of speech.

[3]In this thesis, we follow the convention that canonical pronunciations corresponding to expected phoneme sequences will be indicated within two forward slashes ('/ $\cdots$ /'). To indicate the actual realization (*surface pronunciation*) of the word as might be transcribed by a linguist, we shall use square braces ([$\cdots$]). We shall generally indicate phonemes in the canonical pronunciation using Arpabet symbols; we shall also use Arpabet symbols to indicate phones in the surface pronunciation, possibly modified with diacritics to indicate nasalization, etc. In cases where we transcribe sounds using IPA symbols, we clarify this in the text. A list of Arpabet phonemic symbols along with examples of words containing them in their canonical pronunciation is provided in Appendix C

automatic speech recognizers to recognize. In the 1998 NIST evaluations on broadcast news speech [Pallett et al., 1999] error rates for the various systems were almost twice as high in the case of spontaneous speech relative to the baseline speech corresponding to studio recordings. Further evidence for this fact comes from a more carefully controlled study by Weintraub et al. [1996]. Weintraub et al. collected a corpus of spontaneous two-party conversations using a methodology similar to the one used to collect the Switchboard corpus [Godfrey et al., 1992]: Participants were assigned a topic (e.g., air pollution, buying a car, etc.) and instructed to speak on the topic. After these conversations had been transcribed, participants were instructed *to read transcripts of their own speech* in a conversational style. Weintraub et al. performed word recognition experiments using these two styles of speech and found that the spontaneous conversational speech was recognized with significantly higher error rates (52%) when compared to read transcripts of identical speech (38%). Since the recognizers and speech were identical (including acoustic, pronunciation modeling and language components), the differences between the two error rates can only be attributed to differences in speaking styles.[4]

1.2 Modeling Pronunciation Variation in ASR Systems

It is generally recognized that pronunciation variation is one of the main causes for the comparatively poor performance of automatic speech recognition systems in recognizing spontaneous conversational speech. Evidence for this fact comes from previous work by McAllaster et al. [1998] in which the authors simulate data from acoustic models (by sampling from the corresponding triphone Gaussian distributions according to a distribution

[4]A detailed description of the conventional generative speech recognition model appears in Section 2.1. Informally speaking, the *acoustic model* estimates probability distributions of acoustic features given sub-word states (eg. phones), the *pronunciation model* estimates probability distributions over sub-word state sequences given words and the *language model* estimates probability distributions over word sequences.

model) to evaluate performance in the case where: (a.) the data is simulated according to the dictionary pronunciation available with the recognizer (canonical pronunciation) (b.) the data is simulated according to *phonetic transcriptions of the data* (surface pronunciation). Note that in either case, the recognizer has access to the same dictionary pronunciations. McAllaster et al. found that the error rates are significantly lower when the simulated data corresponds to the dictionary pronunciation (5-10%) compared to the case where data is simulated according to the phone transcripts (40%). Similar reductions in word error rates were also reported in a later 'cheating experiment' by Saraçlar et al. [2000], where it was found that replacing the pronunciation of the word with the observed surface pronunciation (before decoding each utterance) reduced error rates from 47% to 27%. These studies lend support to the view that improved pronunciation modeling can improve ASR performance.

Techniques for dealing with the increased variability in pronunciation observed in conversational speech can be applied at all levels in conventional speech recognition systems - the acoustic model, the pronunciation model as well as the language model [Strik and Cucchiarini, 1999].[5] In fact, Strik and Cucchiarini note that it is likely that a successful ASR system would need to address it at all levels. The dominant approach employed in conventional ASR systems is to use the phonemes as the fundamental unit in the pronunciation of the word. The model assumes that phonemes are strung together to produce the pronunciation of the word leading to the so-called 'beads on a string' approach which has a number of well-known drawbacks [Ostendorf, 1999]. At the acoustic modeling level, co-articulation effects are modeled in such phone-based systems using context dependent models (eg. triphones). Even if the complexities involved in robustly estimating these distributions from

[5]See Section 2.1 for an explanation of these terms.

limited training data are ignored, concerns with this approach remain. Jurafsky et al. [2001] show that triphone-based models capture phone substitution but they do not model phone deletions and insertions well. As Ostendorf [1999] points out, if a phone is deleted in the pronunciation, the triphones chosen in an alternate pronunciation will be different from the observed pronunciation and co-articulation will not be modeled correctly in this case. At the level of pronunciation modeling, the traditional solution is to add pronunciation variants to the lexicon. These pronunciation variants may be either generated using prior linguistic knowledge based on phonological rules [Giachin et al., 1990; Tajchman et al., 1995] or learned from the data [Fosler et al., 1996; Riley et al., 1999] (A comparison of the two approaches is presented in [Wester, 2003]). However, such attempts at accounting for the variability in speech do not address the cause for the variation in speech directly, modeling it instead as the resultant change in the observed surface phonemic representation. Speech is produced as the result of complex and concerted motion of the articulators and it may therefore be desirable to directly model the *cause* for the variability observed in speech. This observation motivates the use of finer-grained representations for modeling pronunciation.

1.3 Modeling Pronunciation Variation at Sub-Phonetic Level: Motivation for Articulatory Feature-based Approaches

The studies briefly described in the previous section bolstered the view that improved pronunciation modeling could benefit ASR. One might ask, however, what is the right level of detail at which we should model the variation [Livescu et al., 2012]: the level of phones (as was done in many of the reviewed studies) or at a sub-phonetic level (as is proposed in

this thesis and many other works)? In a detailed analyis conducted on the phonetically transcribed portion of Switchboard [Greenberg et al., 1996], Saraçlar and Khudanpur [2004] examined the acoustics of phonemes which had been identified as non-canonical by trained linguists and found that in many cases the acoustics of the phoneme were somewhere 'in between' the representations of the acoustics of the canonical and non-canonical phones,[6] suggesting that the change in pronunciations of phonemes is partial: it is not the *entire* phone which is inserted or deleted, but it is modified partially.

The present thesis follows a number of previous studies ([Deng et al., 1997; Richardson et al., 2003; Livescu, 2005; King et al., 2007] inter alia) in modeling pronunciation variation at the sub-phonetic level. Specifically, the pronunciation models used in the thesis are based on models of speech production, wherein the fundamental units of the pronunciation of the word shall be represented in terms of articulatory features.[7] Although we defer a full discussion of the use of articulatory feature-based models until Section 2.3, we end this section with a brief discussion of some of the claims regarding the advantages of such representations that have been made previously in the literature [King et al., 2007]. These arguments can be summarized as follows:

- **Improved modeling of speech variability:** Since variation observed in conversational speech is produced as a result of the complex concerted motion of the speech articulators [Hardcastle, 1985; Browman and Goldstein, 1992; Farnetani and Recasens, 2012], a model that explicitly accounts for this interaction might better explain the resulting variation.

[6] As measured in terms of distance of the acoustic feature vector from the means for the corresponding phones in the acoustic model.

[7] For the purposes of the thesis, we shall use the term *articulatory features* to refer to what are also known as "phonological features" (e.g., IPA features such as manner and place) that can be used to describe phones, as well as abstract representations of the states of the articulators during the production of speech.

- **Natural factorization of the sub-word state:** Sub-phonetic modeling using articulatory features allows monolithic phone-based units to be naturally described in terms of a set of *simpler categories*. In principle, this might allow for simpler classification tasks and thus improved performance.

- **Improved recognition in noisy environments:** Since different aspects of the speech signal may be corrupted differently in the presence of noise [Miller and Nicely, 1955], an articulatory feature-based representation may offer increased robustness to noise [Kirchhoff et al., 2002].

- **Invariance across speakers and languages:** Although the realization of phonemes varies across languages, sub-phonetic representations may be more invariant and thus might offer better cross-language generalization [Stüker et al., 2003a].

1.4 Organization of the Rest of the Thesis

In Chapter 2, we provide some background information. The chapter begins with a brief introduction to aspects of ASR technology, including the main components of an ASR system: the acoustic model, pronunciation model and the language model. In the rest of the chapter, we summarize previous work in ASR that has utilized articulatory feature-based models, and discuss some of their findings. The remainder of the chapter is dedicated to a detailed discussion of the theory of articulatory phonology [Browman and Goldstein, 1992] along with a description of our implementation of an articulatory feature-based pronunciation model, based on previous work [Livescu, 2005], that incorporates aspects of this theory.

In Chapter 3, we implement our articulatory feature-based pronunciation models using conditional random fields (CRFs) [Lafferty et al., 2001] and apply these to the task

of automatically generating articulatory feature transcriptions of speech utterances given their corresponding word transcriptions. We compare our models to previous *generative* dynamic Bayesian network (DBN) [Livescu, 2005] models, finding that our models significantly improve performance in terms of articulatory feature prediction on the Switchboard Transcription Project (STP) dataset [Greenberg et al., 1996].

Chapter 4 describes our models for spoken term detection, which extend previous work [Keshet et al., 2009] in not requiring sub-word state alignments for training examples. We then extend these models to incorporate an articulatory feature-based pronunciation model in Chapter 5 and evaluate these models in the setting of limited data, simulated by selecting subsets of varying size from the Switchboard [Godfrey et al., 1992]. In experimental evaluations, we find that our proposed models outperform baseline hidden Markov Model-based (HMM-based) systems across a range of dataset sizes. We also conduct an analysis of our systems to determine the impact of the articulatory feature-based pronunciation models on capturing pronunciation variation.

In Chapter 6, we adapt the spoken term detection approaches presented in Chapters 4 and 5 to leverage the availability of LVCSR-based spoken term detection systems. In this chapter, we present results of spoken term detection on the IARPA BABEL Cantonese dataset [IAR, 2011]. In experimental results, we find that the acoustic keyword spotting approaches presented in this thesis are competitive against the strong LVCSR-based baseline; combining system outputs with the baseline results in large performance improvements.

We conclude in Chapter 7 with a summary of the results presented in this thesis, and briefly describe possible extensions and future work.

1.5 Contributions of the Thesis

The main contributions of this thesis are the development of discriminative articulatory feature-based pronunciation models, and the application of these models to the task of detecting particular words or phrases in conversational speech. In particular, the contributions of this thesis are:

- **Discriminative Articulatory Feature-based Pronunciation Modeling:** We develop discriminative articulatory feature-based pronunciation models using conditional random fields that explicitly account for the ability of the speech articulators to desynchronize in conversational speech. Additionally, we describe how deterministic task-specific constraints can be exploited to perform exact inference efficiently in these models. The models are applied to the task of generating articulatory feature transcripts of speech utterances given their corresponding word transcripts. In experimental evaluations, we find that the proposed models outperform previously proposed *generative* dynamic Bayesian network models for the task.

- **Discriminative Spoken Term Detection in Low-Resource Settings:** We apply the proposed articulatory feature-based pronunciation models to the task of spoken term detection – detecting whether and where a given word or phrase is spoken in a speech utterance – by extending previous work by Keshet et al. [Keshet et al., 2009]. We evaluate the proposed models in the setting of low-resource conditions, simulated by sampling utterances from the Switchboard dataset [Godfrey et al., 1992], to determine the effectiveness of the proposed approach. In experimental evaluations, we find that the propose approach outperforms baseline hidden Markov model-based (HMM-based) models in a number of settings.

- **Discriminative Spoken Term Detection Leveraging Existing LVCSR-based Systems:** We describe how the proposed discriminative spoken term detection approach can be adapted to leverage existing LVCSR-based spoken term detection systems, if available, in order to improve system performance and running time. In experimental evaluations on the IARPA Babel Cantonese dataset [IAR, 2011], we find that combining the proposed discriminative systems with the baseline results in large improvements.

CHAPTER 2: BACKGROUND

This chapter serves to introduce and review topics that are relevant to the material described in the thesis. This includes both material that serves to explain core ideas as well as a survey of the literature and prior work.

In Section 2.1, we begin by discussing the fundamentals of automatic speech recognition (ASR). The section can be skipped by readers well versed in the basics of ASR technology. We then provide a detailed review of articulatory phonology [Browman and Goldstein, 1992] in Section 2.2. In Section 2.3, we provide a detailed review of previous work that has incorporated articulatory feature-based representations in ASR tasks. Finally, we end this chapter with a brief intuitive description of our implementation of an articulatory feature-based pronunciation model in Section 2.4.

2.1 Automatic Speech Recognition

The current dominant paradigm for ASR is based on probabilistic modeling [Rabiner and Juang, 1993]. Although the field of ASR encompasses a varied set of research problems, we only consider in this section the problem that (arguably) lies at the heart of most ASR research: the task of determining the most likely word sequence corresponding to a given speech utterance.

In what follows, we assume that the speech waveform has been parameterized into suitable acoustic feature vectors corresponding to each frame of speech (i.e., discreted units

of time, typically about 10ms): $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$ where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$.[8] Examples of the acoustic parameterization used ubiquitously in ASR include mel-frequency cepstral coefficients (MFCCs) [Davis and Mermelstein, 1980] and perceptual linear prediction coefficients (PLPs) [Hermansky, 1990]. Although a full review of feature extraction is outside the scope of this thesis, it suffices to say that the acoustic features capture information about the energy present in various frequencies of the speech signal.

With this background, the word recognition problem in ASR can be re-phrased mathematically as follows: Given the input acoustic representation, ($\overline{\mathbf{x}}$), of the speech utterance we seek the most likely word sequence ($\overline{v}^*$) over all possible word sequences ($\overline{v} \in \mathcal{V}^*$), where $\mathcal{V}$ represents the lexicon,[9]

$$\overline{v}^* = \underset{\overline{v}}{\operatorname{argmax}}\, P(\overline{v}|\overline{\mathbf{x}}) \tag{2.1}$$

In the standard generative paradigm, that has dominated ASR for a number of years, Equation 2.1 can be re-written using Bayes theorem as,

$$\overline{v}^* = \underset{\overline{v}}{\operatorname{argmax}}\, P(\overline{v}|\mathbf{x}) \tag{2.2}$$

$$= \underset{\overline{v}}{\operatorname{argmax}}\, \frac{p(\overline{\mathbf{x}}|\overline{v})P(\overline{v})}{p(\overline{\mathbf{x}})} \tag{2.3}$$

$$= \underset{\overline{v}}{\operatorname{argmax}}\, p(\mathbf{x}|\overline{v})P(\overline{v}) \tag{2.4}$$

where, Equation 2.4 follows since the maximization is independent of the prior on the acoustics, $p(\overline{\mathbf{x}})$. Observe that the second term in Equation 2.4 is independent of the acoustics; instead, it computes the probability of any word sequence being uttered by the speaker.

[8] In this thesis, we follow the convention that scalar quantities are written in normal font (e.g., $x \in \mathbb{R}$), vectors are written in bold font (e.g., $\mathbf{x} \in \mathbb{R}^d$), and sequences of vectors are represented using the 'overbar' notation (e.g., $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$).

[9] In this thesis, we use the notation $\mathcal{V}^*$ to denote all sequences of zero or more elements chosen from the set $\mathcal{V}$. When used with lowercase symbols we denote optimal values of that symbol (e.g., $\overline{v}^* = \operatorname{argmax}_{\overline{v}} P(\overline{v}|\overline{\mathbf{x}})$).

The first term, on the other hand, corresponds to the likelihood of observing the speech frames given a particular word sequence, $\overline{v}$.

For all but the simplest of tasks (e.g., small number of words in the lexicon, with many examples per word), estimating acoustic likelihoods, $p(\overline{\mathbf{x}}|\overline{v})$, directly is intractable. Instead, most systems decompose the word into smaller sub-word units (typically, context-independent phonemes (i.e., monophones) or context-dependent phonemes (e.g., triphones)). For example, the word "cat", can be decomposed into three phonetic sounds, /k ae t/, representing the three phonemes in the word. If we generically denote the sequence of sub-word states by $\overline{q}$, we can re-write Equation 2.4 as,

$$\overline{v}^* = \underset{\overline{v}}{\text{argmax}} \sum_{\overline{q}} p(\overline{\mathbf{x}}, \overline{q}|\overline{v}) P(\overline{v}) \tag{2.5}$$

$$= \underset{\overline{v}}{\text{argmax}} \sum_{\overline{q}} p(\overline{\mathbf{x}}|\overline{q}) P(\overline{q}|\overline{v}) P(\overline{v}) \tag{2.6}$$

$$\approx \underset{\overline{v}}{\text{argmax}} \max_{\overline{q}} p(\overline{\mathbf{x}}|\overline{q}) P(\overline{q}|\overline{v}) P(\overline{v}) \tag{2.7}$$

where, for tractability we have assumed that the acoustics $\overline{\mathbf{x}}$ are conditionally independent of the word sequence $\overline{v}$ given the sequence of sub-word units $\overline{q}$, and replaced the summation over phone sequences by a 'max' operation (Viterbi approximation).

Thus, the task of word recognition in ASR can be reduced to the task of estimating the three quantities $p(\overline{\mathbf{x}}|\overline{q})$, $P(\overline{q}|\overline{v})$ and $P(\overline{v})$ in Equation 2.7; techniques for estimating these quantities form important sub-fields in ASR. The quantity represented by the first term, $p(\overline{\mathbf{x}}|\overline{q})$, is known as the *acoustic model* and it represents the likelihood of a set of acoustic vectors being produced as a result of uttering a given sequence of sub-word states. The second term, $P(\overline{q}|\overline{v})$, is known as the pronunciation model. In many ASR systems, estimating this quantity is essentially a dictionary lookup (e.g., P(/k, ae, t/|"cat") $= 1.0$). Finally, the last term in Equation 2.7, $P(\overline{v})$, is known as the language model, and it represents the

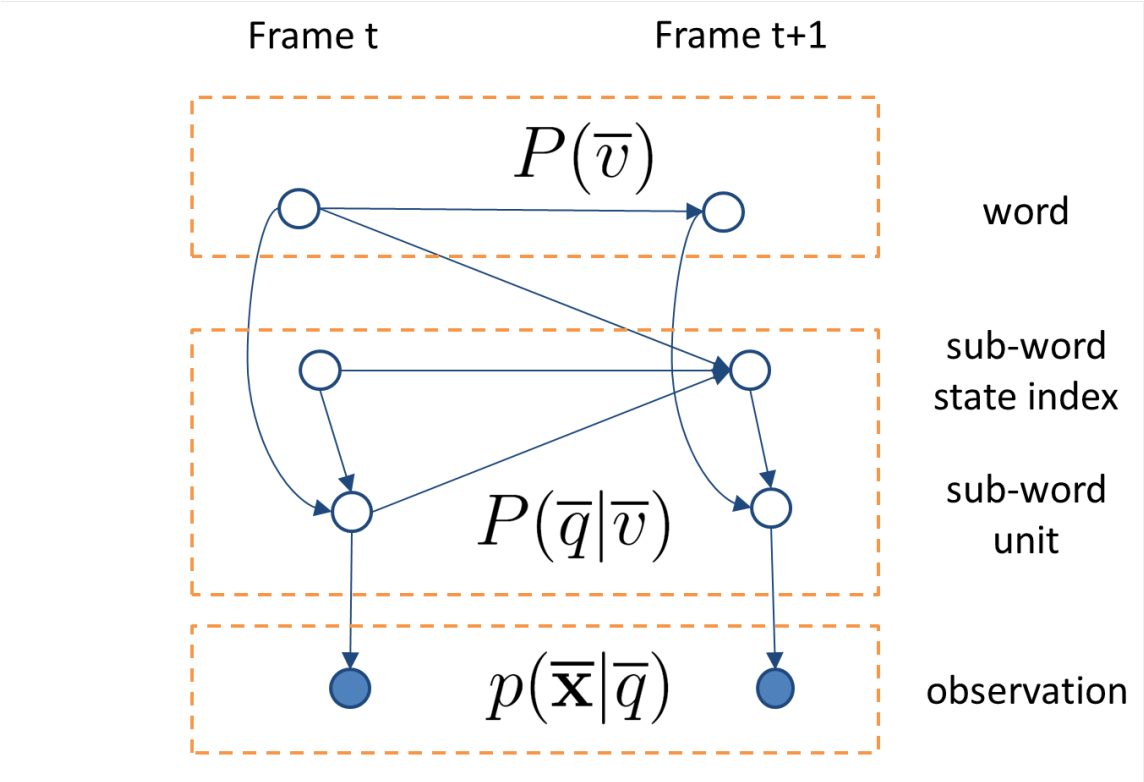

Figure 2.1: The generative ASR process in Equation 2.7, represented as a dynamic Bayesian Network [Zweig, 1998]. The shaded observation nodes indicate that these variables are observed at test time. In this view, a word sequence is first sampled from the language model ($\overline{v} \sim P(\overline{v})$). Given a word sequence, the sequence of sub-word state alignments, representing the pronunciation of the word sequence is sampled from the pronunciation model ($\overline{q} \sim P(\overline{q}|\overline{v})$). Finally, the acoustics are sampled from the acoustic model given the pronunciation in terms of sub-word state alignments ($\overline{\mathbf{x}} \sim p(\overline{\mathbf{x}}|\overline{q})$).

probability that a sequence of words is uttered by the speaker. For those familiar with the notation of graphical models, the *generative* process of speech recognition expressed in Equation 2.7 can be represented as a dynamic Bayesian network (DBN) [Zweig, 1998] as illustrated in Figure 2.1.

In summary, by decomposing the original problem in Equation 2.1 into the form given by Equation 2.7 we can invest modeling effort into improving each of the acoustic, pronunciation and language models with the hope that this will result in improved word recognition.

In the next section we discuss Browman and Goldstein's [1992] theory of articulatory phonology, which is an attempt to account for the pronunciation variation in speech. In the context of Equation 2.7, when incorporated within an ASR system, this work can be thought of as being at the level of the pronunciation model $P(\overline{q}|\overline{v})$.[10] We discuss our specific implementation of this model in Section 2.4.

2.2 Articulatory Phonology

In the theory of articulatory phonology proposed by Browman and Goldstein [1986; 1990; 1992], *gestures* are considered to be the fundamental units of contrast among the words of a language. Gestures are discrete events that occur during the production of a word, corresponding to specific configurations of the vocal tract that are produced as a result of the speech production process. For example, the *bilabial closure gesture*, corresponds to the formation of a complete closure of the vocal tract at the lips and it is present in the production of words that contain bilabial stop consonants (e.g., /b/ and /p/). Thus, the two words 'bad' (pronounced /b ae d/) and 'add' (pronounced /ae d/) are contrasted in the language by the presence or absence, respectively, of the bilabial closure gesture.

Gestures in articulatory phonology are described by a set of related *tract variables* which are illustrated in Figure 2.2. These tract variables specify the degree of constriction

[10]Or perhaps jointly at the level of the acoustic and pronunciation models $P(\overline{\mathbf{x}}, \overline{q}|\overline{v})$, depending on the actual implementation.

(e.g., a complete closure during the production of stop consonants; a *critical* closure producing turbulent airflow during the production of fricatives) and the constriction location (at the lips, tongue tip, tongue body (dorsum), velum (controlling nasalization) and the glottis (controlling voicing)). The states of the tract variables do not correspond directly to individual speech articulators, but instead they correspond to the concerted movements of a set of speech articulators. For example, a bilabial closure gesture (as in the production of /b/ or /p/) is related to the tract variable LA (Lip Aperture) corresponding to the constriction degree at the lips, which in turn is controlled by the motion of the jaw, and the upper and lower lip. The set of tract variables in [Browman and Goldstein, 1992] correspond to the lip (LIPS), the tongue tip (TT), the tongue body (TB) the velum (VEL) and the glottis (GLO).

Since the same articulator might be involved in multiple gestures (and tract variables) that are simultaneously active during the production of a word, the coordination amongst the gestures is specified in terms of *task dynamics* [Saltzman and Munhall, 1989]. In this formalism, the motion of the tract variables in time is described in terms of a second-order dynamical system. In essence, the use of task dynamics allows a set of static gestures involved in the production of a word to be represented dynamically as a *gestural score* [Browman and Goldstein, 1992], that is a representation of the relative timings and overlaps among the tract variables involved in the various gestures. Figure 2.3 illustrates an example of the gestural score for the word 'span' (pronounced /s p ae n/). As can be seen in the figure, the set of gestures in the production of the word – the 'gestural constellations' – overlap in time. These overlaps are produced because of the fact that multiple speech articulators are involved in the production of each of the gestures, and in some cases the same underlying articulator is involved in two simultaneous (in time) gestures.

tract variable		articulators involved
LP	lip protrusion	upper & lower lips, jaw
LA	lip aperture	upper & lower lips, jaw
TTCL	tongue tip constrict location	tongue tip, body, jaw
TTCD	tongue tip constrict degree	tongue tip, body, jaw
TBCL	tongue body constrict location	tongue body, jaw
TBCD	tongue body constrict degree	tongue body, jaw
VEL	velic aperture	velum
GLO	glottal aperture	glottis

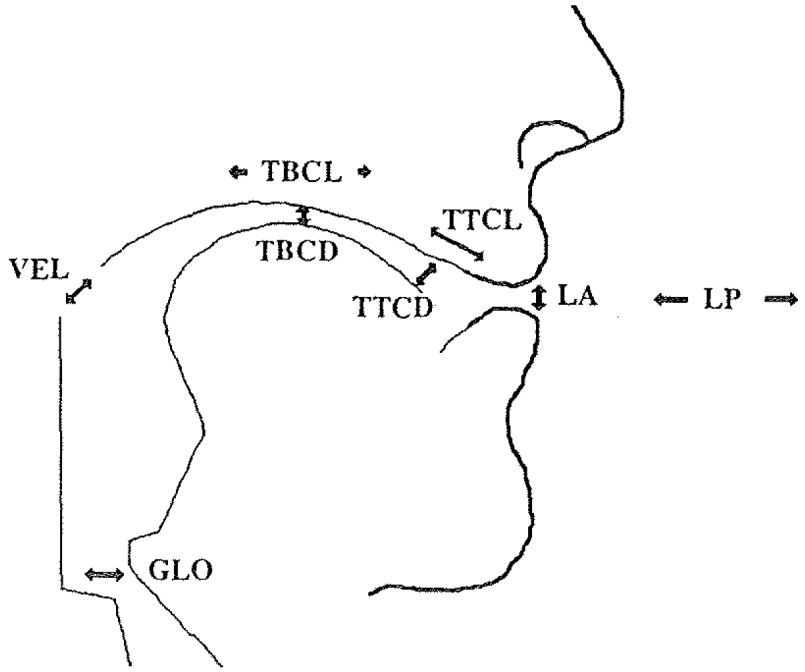

Figure 2.2: The tract variables of articulatory phonology along with the articulators they are associated with, reproduced from [Browman and Goldstein, 1990]. Notice that multiple tract variables share the same underlying articulators. As a result, although gestures are specified independently for each articulator, multiple tract variables can be impacted by the motion of a single articulator.

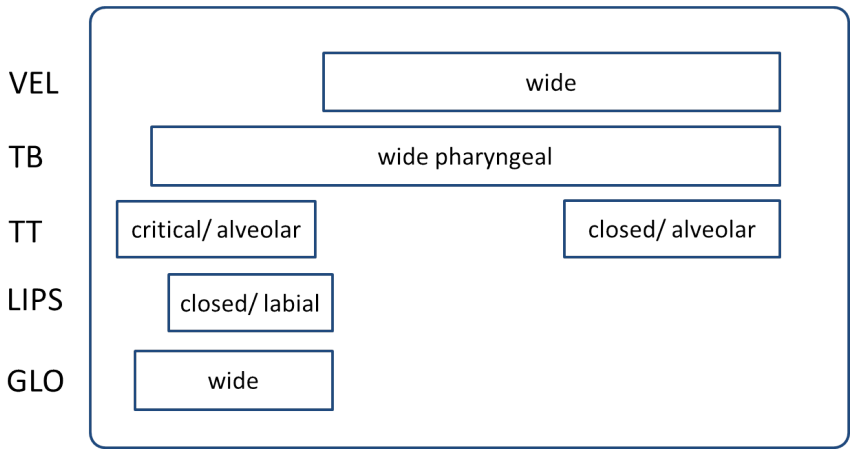

Figure 2.3: An illustration of the gestural score for the word 'span' (pronounced /s p ae n/) adapted from [Browman and Goldstein, 1992]. The horizontal axis indicates a discretized representation of time; the gestures are associated with specific tract variables (see Figure 2.2). The gestural score indicates, for example, that the glottis is wide during the production of the /s/ and /p/ sounds since these are unvoiced, as well as the bilabial closure produced during the stop consonant /p/.

The variation observed in conversational speech can be accounted for through one of two mechanisms within the theory of articulatory phonology: (a.) overlap between gestures corresponding to different tract variables, and (b.) reductions in the *magnitude* of a particular gesture because of the contextual environment in which the gesture is present. In fact, based on linguistic analyses, Browman and Goldstein make the stronger claim that " ... all examples of fluent speech alternations are due to (these two mechanisms) ..." [see Browman and Goldstein, 1990, Section 3.1].

In order to motivate our specific implementation of the pronunciation model based on the theory of articulatory phonology, which is presented in Section 2.4, we describe three examples from the literature that illustrate how these two mechanisms can address some of the variability seen in conversational speech. The discussion of the subject must necessarily be brief; an interested reader is referred to excellent references in the following

papers [Browman and Goldstein, 1986, 1990, 1992] for additional examples and discussions.

The first example that we review discusses how gestural overlaps can help to explain the apparent deletion of a phoneme due to the surrounding context. When spoken in isolation, the final /t/ sound is audible in the word "perfect" (/p er f eh k t/). The same /t/ sounds is inaudible, however, in the phrase "perfect memory" (/p er f eh k t m eh m er iy/) when spoken in a conversational setting. Browman and Goldstein [1990] examined data of subjects producing these words in isolation and in a conversational setting and recorded the movements of the various speech articulators; their data [Browman and Goldstein, 1990] is reproduced in Figure 2.4. The figure shows the vertical displacements of the various speech articulators produced (a.) during the utterance of the words "perfect" and "memory" in isolation, and (b.) during the utterance of the phrase "perfect memory" in a conversational setting. In either case, the tongue tip alveolar closure gesture (marked τ) appears to be present, and its onset is obscured by the presence of the velar closure gesture (marked κ). In the isolated case represented by (a.) the release of the alveolar closure is not completely obscured, and is thus acoustically audible. In the conversational setting represented by (b.), however, even the release of the alveolar closure gesture completely overlaps with the bilabial closure gesture (marked β), and is thus acoustically hidden. Gestural overlaps can thus account for apparent (acoustic) deletions of phonemes although there is evidence that the underlying gesture is present. Similar analysis of gestural overlaps can be used to explain, for example, epenthetic stop insertions (see Figure 2.7), and anticipatory/preservatory rounding.

The second example that we discuss corresponds to Browman and Goldstein's [1992] analysis of the variation seen in consonantal and vocalic gestures: consonantal gestures are generally characterized by shorter duration and higher degrees of constriction relative to

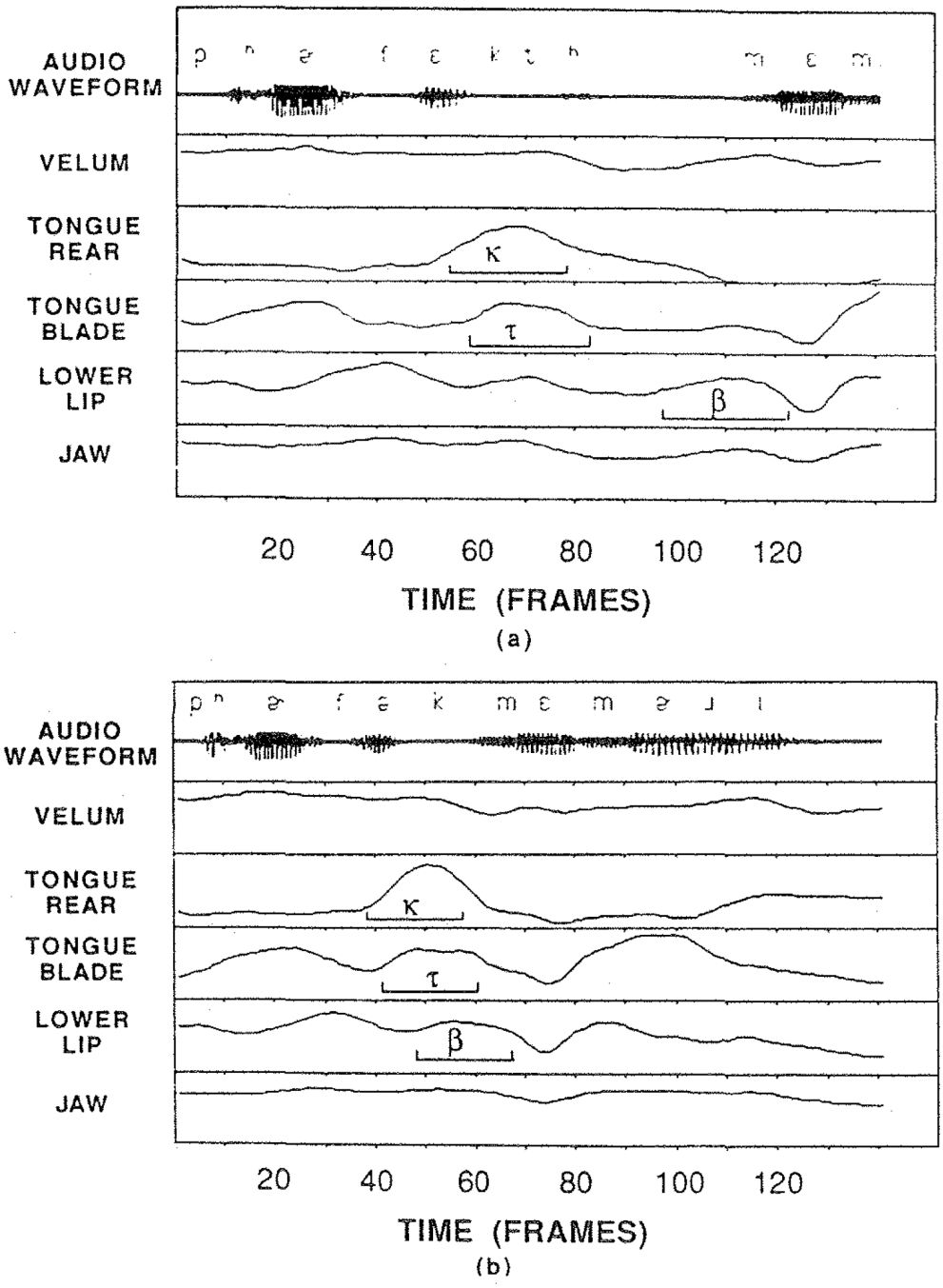

Figure 2.4: Vertical displacements of the various speech articulators produced (a.) during the utterance of the words "perfect" and "memory" in isolation (b.) during the utterance of the phrase "perfect memory" in a conversational setting, reproduced from [Browman and Goldstein, 1990]. The surface phonetic transcription of the audio waveform in the two cases is indicated using IPA symbols: (a.) [pɚfɛkt ˈmɛm...], and (b.) [pɚfɛkˈmɛm...]

.

vowel gestures, and there is significant temporal overlap in these gestures. In cases where the same consonant gesture is produced in the context of different vowel gestures (e.g., (in IPA) /ada/ vs /idi/) the resulting consonantal gesture is affected by the surrounding vowel gestures [Saltzman and Munhall, 1989]. This effect is magnified in cases where the *same articulators and tract variables* are involved in the respective gestures. Thus, in cases such as /ada/ and /idi/, distinct articulators are involved (TT for [d] and TB gestures for the vowels) and the resulting consonantal gesture is relatively invariant. On the other hand, in examples such as (e.g. (in IPA) /aga/ vs. /igi/), both the consonant and the vowel involve gestures corresponding to the same tract variable (TB). In such instances, the actual vocal tract structures produced (but not the degree of constrictions, i.e. not gestural magnitudes) show more variation since the same speech articulators are involved in the consonantal and vocalic gestures.

Our final example concerns some instances of variation in conversational speech such as lenitions of stop consonants to the corresponding fricatives [Browman and Goldstein, 1990]. For example, instances where phrases such as "must be" (IPA canonical: /ˈmʌst bi/) are produced instead as (IPA: [ˈmʌsβi]). Such instances can be explained to be produced as a result of reductions in gestural magnitude (i.e. a *complete* closure being instead replaced by a *critical* closure). Such reductions are likely to be more common in fast conversational speech rather than carefully articulated speech.

The primary motivation of the work presented in this thesis is to address the pronunciation variation in observed in conversational speech, with the goal of improving ASR technologies. Towards this end, in Section 2.4 we describe our implementation of a pronunciation model for ASR that is based on the theory of Articulatory Phonology, which is

based on previous work by Livescu and colleagues [Livescu and Glass, 2004a,b; Livescu, 2005; Livescu et al., 2007a].

2.3 Articulatory Feature-based Models in ASR

Following the brief introduction to articulatory phonology presented in the previous chapter, this section serves as a review of the literature wherein we highlight previous studies that have applied articulatory feature-based models[11] in ASR. This section also serves to motivate the articulatory feature-based pronunciation models that we study as part of the thesis.[12]

As was briefly mentioned in Chapter 1, numerous arguments have been made in the literature advocating the use of articulatory feature-based approaches in ASR in order to overcome some of the limitations of phone-based pronunciation modeling [Ostendorf, 1999]. These arguments can essentially be summarized as follows [Rose et al., 1996; King et al., 2007; Livescu et al., 2012]: (a.) the ability to (automatically) obtain information about articulation from the speech signal would aid further scientific research in fields such as speech science, linguistics and ASR (b.) ASR models that encode *apriori* knowledge of speech production might better capture the variability in the speech signal (c.) articulatory feature-based representations represent a natural factorization of the phonetic state space,

[11] For the purposes of this thesis, we use the term "articulatory feature-based models" to refer to models that have been variously referred to as "speech production models", "phonological feature models", and "gestural models" in the literature.

[12] In the interest of brevity, we do not describe systems that only seek to estimate articulatory features from the data (i.e. articulatory feature classification/recognition or articulatory inversion), but instead focus on studies where articulatory features are used as part of a larger task such as word recognition. Systems that focus on extracting articulatory features from the speech signal are discussed in Section 3.2.

and might lead to simpler classification problems (d.) articulatory feature-based representations might generalize better across languages (e.) articulatory feature-based representations might offer increased robustness in the presence of noise. The various studies that we describe below support these hypotheses to various degrees.

In a number of previous approaches, articulatory feature classifiers of phonological features such as manner of articulation, place of articulation, voicing etc. have been employed for phone and word recognition tasks. In such systems, the articulatory feature classifiers serve as an additional source of information, which in some of the systems is complementary to a standard phone-based system. These systems, however, do not incorporate an explicit model of articulatory feature asynchrony or gestural overlaps as described in Section 2.2. Kirchoff et al. [Kirchhoff, 1999; Kirchhoff et al., 2002] use multilayer perceptron-based (MLP-based) articulatory feature classifiers in a hybrid speech recognition system [Morgan and Bourlard, 1995] leading to improved word recognition accuracy in the case of speech recognition in noise. Metze and Waibel [2002] train Gaussian mixture models for each of a set of articulatory features, which are then combined with standard context-dependent phone-based acoustic models to get a (weighted) combined acoustic-likelihood. The resulting system was found to improve word recognition performance when evaluated on a broadcast news task. Stüker et al. [2003a; 2003b] train Gaussian Mixture Model-based articulatory feature classifiers on multiple languages and find that incorporating information from articulatory feature classifiers trained on data from other languages helps improve word recognition accuracy for the target language. Multilayer perceptron-based detectors of phones and phonological features have also been used within the framework of a conditional random field-based ASR for phone and word recognition tasks [Morris and Fosler-Lussier, 2008; Morris, 2010].

In contrast with the systems described in the previous paragraph, a number of previous works have also incorporated an explicit model of articulatory feature asynchrony or "feature spreading" in order to directly account for pronunciation variability. Some of the earliest work incorporating articulatory feature-based pronunciation models within probabilistic ASR systems includes work by Deng and colleagues [Deng and Sun, 1994; Deng et al., 1995, 1997]. In these hidden Markov model (HMM)-based systems, the HMM states are modeled as vectors of articulatory feature variables representing configurations of the various articulators such as the lip, tongue, velum and glottis. The expected transcription in terms of the features is allowed to "expand" as a result of *feature spreading*, with the amount of spreading controlled through various rules and constraints, thus modeling gestural overlaps. Deng et al. observed improved performance on the TIMIT phone classification task using the articulatory feature-based approach. Erler and Freeman [1996] describe a similar system where HMM states are represented by configurations of articulatory variables based on articulatory phonology [Browman and Goldstein, 1992], with the variables being treated as ordered categorical variables. This representation in terms of ordered categorical variables allows for the enforcing of smoothness amongst the articulators. This approach was more recently extended by Richardson et al. [2003] to use diphone-based units, which in combination with the baseline was shown to be effective in improving performance in both clean and noisy speech.

Mitra et al. [2009; 2011a; 2011b] train multilayer perceptrons to estimate (continuous-valued) tract variables of articulatory phonology on synthetically generated training data. Once trained, the MLPs can be used to generate tract variable trajectories for real speech utterances. These estimated tract variable trajectories are used as acoustic feature vectors – either as standalone features or in combination with MFCCs – in a hidden Markov

model [Mitra et al., 2009] or a dynamic Bayesian network [Mitra et al., 2011a]. In experimental evaluations, Mitra et al. find that their proposed approach improves word recognition performance in noisy conditions, although word recognition performance did not improve in clean speech. It should be noted that similar observations have been made previously by other authors as well. For example, Wrench and Richmond [2000] conduct experiments using ground truth articulatory trajectories computed using electormagnetic articulography [Wrench, 2001] used either directly or in combination with MFCCs in a standard GMM-HMM system. In their experiments (in clean acoustic conditions) they find that although the MFCC baseline outperforms the articulatory feature baseline, combining the two features results in improved performance over the MFCC baseline. However, no such improvements were found when the articulatory trajectories were estimated directly from the speech signal [Wrench and Richmond, 2000; Frankel et al., 2000; Frankel and King, 2001].

The models of pronunciation used in this thesis are based on models previously proposed by Livescu and colleagues [2004a; 2004b; 2005; 2007a] in a *generative* framework based on dynamic Bayesian networks. In these models, articulatory feature streams (based on the tract variables of articulatory phonology) are modeled as separate streams that are loosely synchronized.[13] Pronunciation variation in these systems is modeled in terms of asynchrony between adjacent feature streams (modeling gestural overlaps) and substitution of one articulatory feature value for another (modeling reductions of gestural magnitudes). In lexical access experiments – the task of predicting word identity given surface phonetic sequences – the proposed approach was shown to outperform a phone-based pronunciation model with phonological rules modeling pronunciation variation [Livescu and Glass,

[13]The models are described in greater detail in Section 2.4

2004a,b]. The addition of context dependent articulatory feature substitution resulted in further improvements [Jyothi et al., 2011]. There have also been recent discriminative extensions of this work using a large-margin algorithm employing a large number of features including some based on articulatory feature streams [Tang et al., 2012] as well as a technique for discriminatively re-weighting arcs of a finite state tranducer (FST) representing the original DBN [Jyothi et al., 2012]. The DBN-based articulatory feature pronunciation models were subsequently incorporated as part of an end-to-end speech recognizer [Livescu et al., 2007a]. However, the results of that study on the SVitchboard dataset [King et al., 2005] indicated that the feature-based models did not outperform baseline monophone-based systems.

Finally, we note that there has been some recent work at building computational models of articulatory phonology that has attempted to more directly and faithfully capture some aspects of the theory,[14] such as task dynamics [Zhuang et al., 2008, 2009; Hu et al., 2010]. In these works, the trajectories of the various tract variables are simulated, and attempts are made at extracting the gestural activation scores from these trajectories. However, since these models have only been evaluated on synthetic data, it is hard to compare them to the other approaches discussed in this section.

2.4 A Pronunciation Model Inspired by the Theory of Articulatory Phonology

The models proposed in this thesis, utilize a pronunciation model based on the theory of Articulatory Phonology [Browman and Goldstein, 1992]; these models are based on those developed previously by Livescu and Glass [Livescu and Glass, 2004a,b; Livescu, 2005] with some modifications, which we outline in subsequent sections.

[14] As was the case in work by Mitra et al. [2009; 2011a; 2011b] as well.

We use the word "sense" as a running example throughout this thesis in order to illustrate the features of our model. Since our end goal is the application of the proposed pronunciation model within ASR technology, we begin with the assumption that we have a standard phone-based pronunciation dictionary that lists the canonical pronunciation of words in terms of phones. The canonical pronunciation of "sense", found in a machine-readable dictionary such as CMUdict [Weide, 2007], appears as /s eh n s/. In a conventional speech recognizer [Young et al., 2002], the pronunciation of the word would be represented in terms of models corresponding to these constituent phonemes (context-independent modeling) or using context-dependent phoneme models (e.g., triphones).

Pronunciations represented as sequences of articulatory feature targets

In our model, we shall instead represent the pronunciations of words in terms of a sequence of articulatory feature targets for a set of articulatory feature variables that correspond to the tract variables of articulatory phonology. Specifically, these represent the constriction degrees and positions of the lips, the tongue tip, the tongue body and the state of the velum and glottis. In our work, we assume that each phoneme can be deterministically mapped to a set of articulatory feature targets one for each articulatory feature.[15] Thus, the pronunciation of a word can be represented as a matrix of feature values obtained by deterministically mapping each of the phonemes in its canonical pronunciation to their corresponding articulatory features values. This is illustrated for the word, "sense" in Figure 2.5. Notice in particular that the sequence of phone targets is not part of the representation of the word, but is provided to indicate the canonical pronunciation of the word.

[15] In our work, we use the mapping outlined in [Livescu, 2005] to map (American) English phonemes to articulatory features. In later work, this mapping is modified slightly when applied to our experiments on spoken term detection.

Phone	s	eh	n	s
VEL	non-nasal	non-nasal	nasal	non-nasal
GLO	wide	critical	critical	Wide
TB	uvular/ medium	palatal/ medium	uvular/ medium	uvular/ medium
TT	alveolar/ critical	alveolar/ medium	alveolar/ critical	alveolar/ critical
LIPS	wide/ labial	wide/ labial	wide/ labial	wide/ labial

Figure 2.5: Representation of the canonical pronunciation of "sense" in terms of sequences of articulatory feature targets for each of the articulatory feature streams. The notation (x:y), which is used in describing the TT, TB, and LIPS streams is used to differentiate the position (x) and the constriction degree (y) corresponding to the articulator. Note that the 'Phone' stream indicated in the figure is only provided to indicate the *surface* pronunciation in terms of phones corresponding to the articulatory feature values and is not included in the representation of the pronunciation.

As can be seen in Figure 2.5, the word "sense" has four articulatory feature targets for each stream (since it has four phonemes in its canonical pronunciation). For example, the sequence of articulatory targets for the velum stream (VEL) are [non-nasal, non-nasal, nasal, non-nasal] since only the third phoneme (/n/) corresponds to nasal sound. In terms of articulatory phonology, this corresponds to positing a *wide velum gesture* during the production of the third unit in the pronunciation of the word. It should be noted however, that the pronunciation representation in Figure 2.5 is distinct from a gestural score (Figure 2.3) in two ways: (a.) Firstly, in our model, the pronunciation representation is fully specified, with an articulatory feature value indicated for each unit of the pronunciation, unlike the extremely underspecified representation in the gestural score. Secondly, and more critically, the pronunciation representation in Figure 2.5 does not indicate relative timing of

the articulatory targets, but only represents that sequences of targets that must be realized during the production of the word.

Asynchronous evolution of articulatory streams

We assume that during the production of the word, each stream passes through the series of targets specified in the pronunciation of the word. Thus, given a speech utterance corresponding to the word, each element in the matrix representation of the pronunciation in Figure 2.5, is associated with a start and end time in the utterance. In other words, each stream can transition independently and asynchronously from one articulatory target to the next at each speech frame. Thus, the model endows an extent in time to these articulatory feature targets, which allows us to model the notion of gestural overlaps in the theory of articulatory phonology. Further, in order to prevent completely implausible gestural overlaps from being hypothesized by the system, we impose constraints on the maximum amount of asynchrony (in terms of state overlaps) permitted in the system.

Continuing with our running example, consider an instance of the word 'sense' where each of the feature streams is completely synchronized with respect to the others. In this case, the resulting *surface pronunciation* corresponds, by design, to the canonical pronunciation as illustrated in Figure 2.6. However, if the streams are desynchronized with respect to each other, then the resulting surface pronunciation may differ from the canonical pronunciation. Such a mechanism can account for the variation that is produced as a result of gestural overlaps. For example, in the phonetically transcribed portion of the Switchboard dataset [Godfrey et al., 1992] – the Switchboard Transcription Project (STP) data [Greenberg et al., 1996] – a variant pronunciation of 'sense' is observed as /s eh^{n} n t s/ which contains a nasalized vowel and an inserted epenthetic stop between the nasal /n/ and the fricative /s/. This particular variant can be explained by the model if we hypothesize it to

VEL	non-nasal	non-nasal	nasal	non-nasal
GLO	wide	critical	critical	wide
TB	uvular/medium	palatal/medium	uvular/medium	uvular/medium
TT	alveolar/ critical	alveolar/ medium	alveolar/closed	alveolar/critical
LIPS	wide/labial	wide/ labial	wide/labial	wide/labial
Phone	s	eh	n	s

Figure 2.6: Example showing how the canonical pronunciation of 'sense' is hypothesized when all of the articulatory feature streams are synchronized with respect to each other. The horizontal axis represents the evolution of time. The 'Phone' stream represents the surface pronunciation in terms of phonemes corresponding to the combination of articulatory features values at a given frame and is not part of the articulatory feature-based pronunciation model.

be the result of a desynchronization of the velum and glottis streams from the other streams as illustrated in Figure 2.7.

2.5 Concluding Remarks

The description in Section 2.4 describes the main aspects of the pronunciation model utilized in this thesis. In particular, we note that by modeling speech in terms of loosely-coupled articulatory feature streams, we can account for pronunciation variation produced as a result of gestural overlaps, which are ubiquitous in conversational speech [Farnetani and Recasens, 2012]. However, as we noted in Section 2.2, the theory of articulatory phonology accounts for pronunciation variation through two mechanisms: gestural overlap and diminished gestural magnitudes. Unlike the pronunciation model of Livescu [2005],

VEL	non-nasal	non-nasal	nasal	non-nasal	
GLO	wide	critical	critical	wide	
TB	uvular/medium	palatal/medium	uvular/medium	uvular/medium	
TT	alveolar/ critical	alveolar/ medium	alveolar/closed	alveolar/critical	
LIPS	wide/labial	wide/ labial	wide/labial	wide/labial	
Phone	s	eh^n	n	t	s

Figure 2.7: Example showing how variant pronunciation for *sense* (epenthetic stop insertion and nasalization of vowel) can be produced when the velum and glottis streams desynchronize from the other streams. The horizontal axis represents the evolution of time. Note that the sequence of articulatory feature values in this example is identical to those appearing in Figure 2.6; the example differs only in terms of the relative transitions between the feature streams. The 'Phone' stream represents the surface pronunciation in terms of phonemes corresponding to the combination of articulatory features values at a given frame and is not part of the articulatory feature-based pronunciation model.

wherein reductions in gestural magnitudes are modeled through articulatory feature substitution, the models used in this thesis do not explicitly model such reduction effects. We briefly justify this design choice in this section.

The primary reason that articulatory feature substitution is not incorporated into our pronunciation models is because of computational considerations: models with articulatory feature substitution, such as the ones presented in [Livescu, 2005], require significantly larger amounts of time in order to perform inference, which is prohibitive when working with large datasets. Secondly, the distinction between the *target feature values* (modeled in our experiments), and the *surface feature values* (modeled in Livescu [2005], but not in our work), is arguably more relevant only in our first set of experiments where we attempt to predict the actual articulatory feature values corresponding to a speech utterance. In

later experiments presented in Chapter 5, we treat the articulatory features as latent variables; in this case, articulatory feature reduction effects are modeled implicitly through the use of feature functions constructed from multilayer perceptron classifiers of phones and articulatory features.

2.6 Summary

In this section, we described the fundamentals of automatic speech recognition and the theory of articulatory phonology [Browman and Goldstein, 1992]. We then described previous systems that have used articulatory feature-based models for ASR. Finally, we briefly described our implementation of a pronunciation model based on previous work [Livescu, 2005].

CHAPTER 3: ARTICULATORY FEATURE FORCED TRANSCRIPTION USING CONDITIONAL RANDOM FIELDS

In this chapter, we describe a set of experiments that are aimed at automatically extracting articulatory feature (AF) targets from speech utterances, given their corresponding word transcriptions.[16] We use the term "articulatory feature forced transcription" to describe such a task, drawing analogy to the task of (phonetic) forced transcription – deriving phonetic labels for speech utterances given their corresponding word transcriptions – a task that finds numerous applications in ASR technology (e.g., in order to derive labels for training multilayer perceptrons for neural network acoustic modeling [Morgan and Bourlard, 1995]).

In this chapter:

- We propose a Conditional Random Field-based (CRF-based) model for articulatory feature forced alignment in Section 3.5 that incorporates the articulatory feature-based pronunciation model that we discussed in Section 2.4. As shall be demonstrated in Section 3.5.2, the models that we propose admit *extremely efficient and exact* algorithms for inference when the deterministic task-specific constraints are exploited.

- In pilot experiments presented in Section 3.6, we demonstrate the effectiveness of the proposed approach over baseline dynamic Bayesian network (DBN) models.

[16] A version of the work described in this chapter has previously appeared in [Prabhavalkar et al., 2011].

We begin in Section 3.1 by motivating the problem and discuss previous work on extracting aspects of articulation from speech utterances in Section 3.2. We then proceed to formally define the problem that we shall solve, and introduce relevant notation in Section 3.3. In Section 3.4, we describe a dynamic Bayesian network-based (DBN) model [Murphy, 2002] for the solution of the problem, based on previous work [Livescu and Glass, 2004a,b; Livescu, 2005]. In Section 3.5 we describe our proposed conditional random field-based (CRF-based) [Lafferty et al., 2001] model for the task. The proposed CRF-based model retains the factorization of the DBN model, and is essentially a *discriminative* version of the *generative* DBN model. In Section 3.5.2, we describe how inference can be performed efficiently in the model by exploiting task-specific constraints relevant to the AF forced-transcription problem. We report the results of experimental evaluations conducted on a subset of the Switchboard Transcription Project (STP) data [Greenberg et al., 1996] in Section 3.6 where we observe that the proposed discriminative CRF-based models offer superior classification performance over the generatively trained DBN baseline.

3.1 Motivation

The CRF-based models for forced transcription that we study in this chapter have two primary motivations. The first seeks to address a major difficulty associated with the development of articulatory feature-based models: the lack of speech data transcribed at the articulatory level. To the best of my knowledge, only three such datasets have been widely used in the community: the MOCHA-TIMIT database of electromagnetic articulography data (EMA) [Wrench, 2001], the University of Wisconsin X-Ray Microbeam Data [Westbury, 1994], and the MRI-TIMIT database of magnetic resonance imaging (MRI) data [Narayanan et al., 2011]. Although the availability of such datasets is extremely

useful, they have characteristics that may make them unsuitable for certain kinds of research. For example, they may contain a significant amount of noise, which is an artifact of the collection process. Additionally, in some cases, certain articulatory measurements may not be available due to the modalities of the collection process. Finally, such continuous measurements are extremely subject-dependent; obtaining a speaker-independent representation is a non-trivial task, although this has been done in some previous studies [Sun et al., 2000; Stephenson et al., 2000; Frankel, 2003].

An alternative approach to obtaining articulatory feature labels has been manual transcription from the speech signal [Livescu et al., 2007b] or mapping from phonetic labels [Kirchhoff, 1999; King and Taylor, 2000; Stüker et al., 2003a; Livescu et al., 2007a]. However, the process of manual transcription of articulatory feature or (detailed) phonetic labels is an extremely time-consuming and difficult process [Greenberg et al., 1996]. Although mapping from phonetic labels is also far from ideal, since these labels may not correspond exactly to the ground truth articulatory features, this has been done in a number of previous studies (including the experiments in this chapter).

The work presented in this chapter has two primary motivations. First and foremost, the ability to automatically derive articulatory feature labels by utilizing available word transcriptions, would help support further research on articulatory features in both linguistics and ASR. Secondly, the task serves as a convenient setting in which to evaluate the effectiveness of the proposed discriminative articulatory feature-based pronunciation models and is thus a stepping stone towards the relatively more complex models of spoken term detection models that we explore in the Chapter 5.

3.2 Background

The task of learning aspects of articulation (e.g., vocal tract configurations, continuous articulator trajectories, articulatory feature categories) has been extensively studied in the past. In early pioneering work on articulatory-to-acoustic inversion, where the goal is to determine the vocal tract area function,[17] Atal et al. [1978] demonstrated that the mapping from acoustics to vocal tract configurations is many-to-one; multiple vocal tract configurations can result in the same acoustic signal. However, this ambiguity can be substantially reduced by enforcing smoothness constraints on the articulator movements [Schroeter and Sondhi, 1994].

Subsequently, a number of studies have attempted to directly recover continuous-valued articulator positions from human speech data directly and have achieved positive results. These attempts have included the use of neural networks [Papcun et al., 1992; Frankel et al., 2000; Richmond, 2001; Mitra et al., 2011b], hidden Markov models (HMMs) [Hiroya and Honda, 2004], analysis-by-synthesis [McGowan, 1994], approaches based on acoustic-to-articulatory codebooks [Hodgen et al., 1996; Suzuki et al., 1998], and using a generalized smoothness criterion during decoding [Ghosh and Narayanan, 2010, 2011]. The average RMS error in reconstructing articulatory trajectories are as low as 0.01mm – 2mm in some of the studies.[18]

Atempts have also been made at recovering discretized articulatory feature categories from the speech signal. These include neural networks [Kirchhoff, 1999; King and Taylor, 2000; Metze and Waibel, 2002; Frankel et al., 2007a; Mitra et al., 2011c], nearest

[17] The area function is typically computed at a few points along the vocal tract, say 20 for example.

[18] A word of caution is prudent here. The RMS error figures indicated here relate to reconstruction errors on different datasets and cannot be directly compared to determine the best performing method. The fact that the reconstruction errors are so low, does however indicate that it is possible to recover articulatory positions accurately from the acoustics.

neighbor-based approaches [Næss et al., 2011], dynamic Bayesian networks [Frankel and King, 2005; Frankel et al., 2007b] and Gaussian mixture models [Stüker et al., 2003b; Jou et al., 2006].

3.3 Notation and Preliminaries

Before formulating the problem formally, we begin by introducing our notation. We assume that we are provided with a speech waveform corresponding to the pronunciation of a single word excised from an entire utterance of conversational speech along with the identity of the corresponding word.[19] We assume that the waveform is parameterized into acoustic feature vectors (e.g., PLPs), $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, where T is the number of frames in the speech utterance and $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ is a feature vector for frame t. Given an utterance $\overline{\mathbf{x}}$ corresponding to the word $\overline{v}$, we denote by $|\overline{v}|$ the number of phones in the canonical pronunciation of $\overline{v}$.

As we described briefly in Section 2.4, we model the pronunciation of the word, $\overline{v}$, as a matrix of articulatory feature targets corresponding to K articulatory feature streams.[20] We assume that we have a mapping from phones in the canonical pronunciation to the corresponding articulatory feature targets for each of the streams. We denote the sequence of articulatory feature targets for stream i as $(\sigma^i_1, \sigma^i_2, \cdots, \sigma^i_{|\overline{v}|})$. Note that the number of articulatory feature targets for each stream is equal to the number of units in the pronunciation of the word.

Formally, the problem of articulatory feature forced-transcription is stated as follows: Given a sequence of parameterized acoustic feature vectors, $\overline{\mathbf{x}}$, and the corresponding word

[19] Our methods can be straightforwardly extended to multi-word sequences, thus allowing us to model cross-word gestural overlaps. In the present work, however, we only consider intra-word gestural overlaps.

[20] In experiments, we assume that lip features form a fully synchronized "bundle", as do all tongue features and the pair (glottis, velum), so $K = 3$.

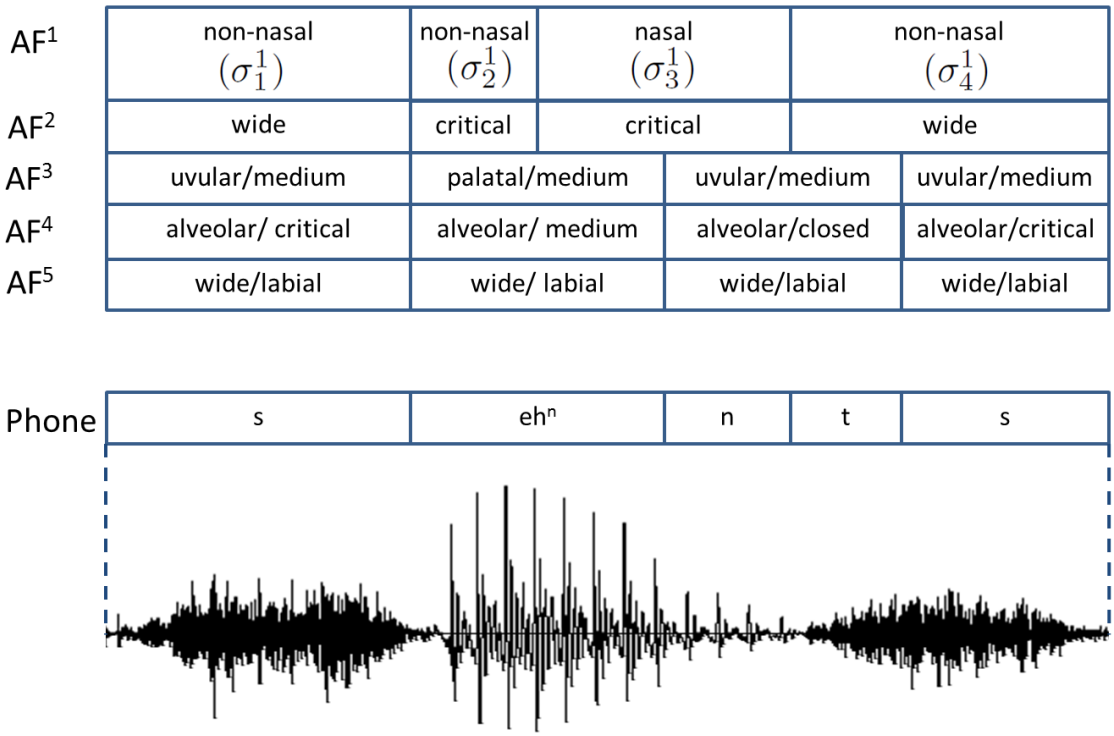

Figure 3.1: Example illustrating the notation used in our experiments on articulatory feature forced-transcription presented in this chapter. In this example, the word $\overline{v}$ = 'sense', with canonical pronunciation /s eh n s/. The corresponding representation of the pronunciation in terms of articulatory feature targets and the corresponding most likely articulatory feature segmentation is illustrated in the figure. The 'Phone' stream indicates the resultant *surface pronunciation* corresponding to the joint configuration of articulatory features at each frame.

transcription, $\overline{v}$, we seek to estimate the value of the articulatory features for each of the streams, $1 \leq i \leq K$, which we denote by $(\overline{\text{AF}}^1, \overline{\text{AF}}^2, \cdots, \overline{\text{AF}}^K)$. Mathematically, we seek to estimate,

$$\overline{\text{AF}}^{1*}, \overline{\text{AF}}^{2*}, \cdots, \overline{\text{AF}}^{K*} = \underset{\overline{\text{AF}}^1, \overline{\text{AF}}^2, \cdots, \overline{\text{AF}}^K}{\text{argmax}} P(\overline{\text{AF}}^1, \overline{\text{AF}}^2, \cdots, \overline{\text{AF}}^K | \overline{v}, \overline{\mathbf{x}}) \tag{3.1}$$

Our notation is illustrated in Figure 3.1.

3.4 Dynamic Bayesian Network-based Model for Articulatory Feature Forced-Transcription

In this section, we describe in great detail a implementation of a dynamic Bayesian network (DBN) [Murphy, 2002] model that formalizes the intuitions of the pronunciation model that we described in Section 2.4. A DBN for the task of AF forced-transcription, based on the models previously proposed by Livescu and Glass [2004a; 2004b] and in Livescu's PhD Thesis [2005], is presented in Figure 3.2. In contrast with the models proposed in those works, we consider a model that captures feature asynchrony, but does not take into account substitution of articulatory features. In other words, we assume that each articulatory feature reaches the value that is expected in the pronunciation of the word at some point during its trajectory. As we mentioned in Section 2.4, the primary justification for this choice lies in computational considerations; in pilot experiments we found inference in the model with substitution to be prohibitively slow.

Asynchronously Evolving Feature Streams

The DBN model in Figure 3.2 can be understood by first examining the role of the sub-word state variables (Sub-word State$_t^i$) corresponding to each of the articulatory feature streams. As the various articulatory feature streams evolve asynchronously, the sub-word state variables represent indices into the pronunciation of the word corresponding to the particular feature stream at the given frame of speech. Thus, each of the sub-word state variables, (Sub-word State$_t^i$), can take a value corresponding to one of the $|\overline{v}|$ articulatory targets for that stream: $1 \leq$ Sub-word State$_t^i \leq |\overline{v}|$.

Using the example of *sense* illustrated in figure 2.6, the sub-word state 1 for the glottis stream corresponds to the phone /s/ and would thus have as its corresponding value the

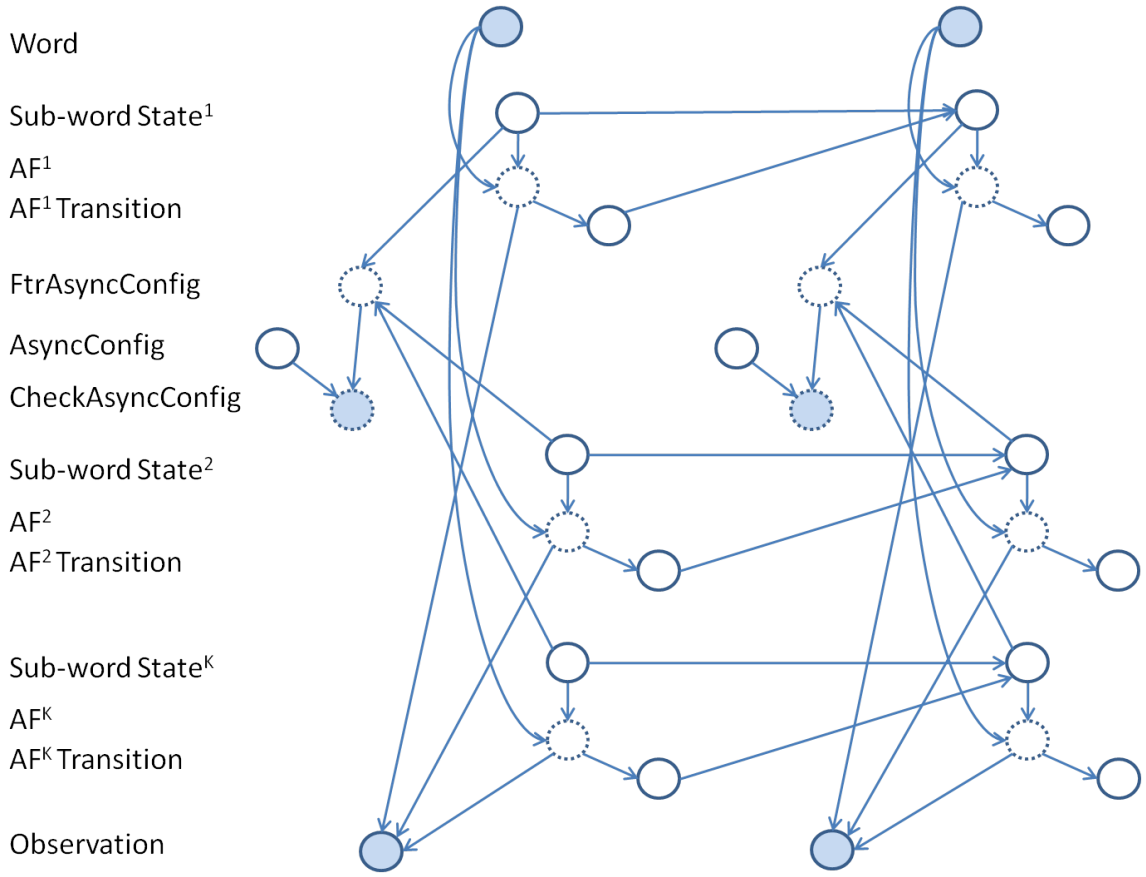

Figure 3.2: Baseline DBN model for articulatory feature alignment based on the work of Livescu and Glass [Livescu and Glass, 2004a,b; Livescu, 2005]. Variables whose values are observed are represented as filled circles (representing the acoustics and word identity); hidden variables are represented as empty circles. Variables whose values are determined deterministically, given the values of their parents, appear as dashed circles.

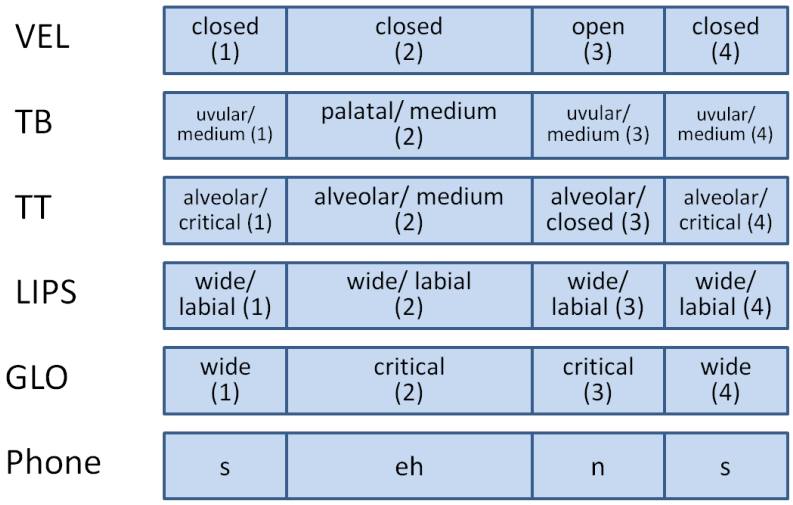

Figure 3.3: Example showing how canonical pronunciation for *sense* is produced when transitions for all feature streams are completely synchronized. In this figure, we have additionally indicated the values that the sub-word state index variables (Sub-word State^i) would take corresponding to each unit in the pronunciation of the word in parentheses.

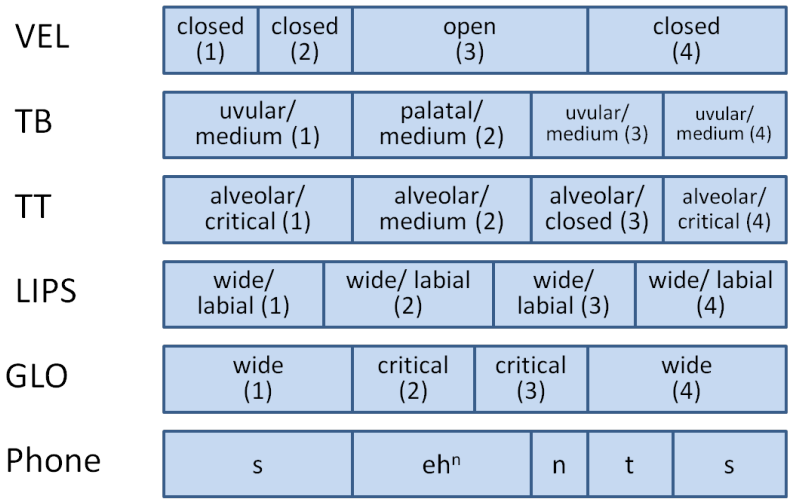

Figure 3.4: Example showing how variant pronunciation for *sense* (t-insertion) can be produced when feature streams desynchronize. In this figure, we have additionally indicated the values that the sub-word state index variables (Sub-word State^i) would take corresponding to each unit in the pronunciation of the word in parentheses.

label *wide* indicating that the sound is unvoiced. Sub-word state 2 for the glottis stream, on the other hand, corresponds to the phone /eh/ which is voiced, and hence the corresponding label for the glottis stream in this state would correspond to *critical*. This is illustrated in Figures 3.3 and 3.4.

Since the model does not allow articulatory feature substitution and the identity of the word is available at test time (see Equation 3.1), we can *deterministically* determine the value of the of the *expected* or *target* articulatory feature for that stream by examining the canonical pronunciation of the word and the corresponding sub-word state variable. Even without a model of articulatory feature substitution, the model is capable of accounting for pronunciation variation that might arise due to feature asynchrony including preservatory and anticipatory rounding, nasalization and epenthetic stop insertions.

Asynchrony constraints

The model of asynchrony is refined further by applying additional constraints: For each pair (i, j) of articulatory features, we define the degree of asynchrony between the two streams $(d_t^{i,j})$ at time-frame t as the difference of the sub-word state indices corresponding to the two streams at that time-frame:

$$d_t^{i,j} = \text{Sub-word State}_t^i - \text{Sub-word State}_t^j \tag{3.2}$$

Thus, if two streams are synchronized with respect to each other, the degree of asynchrony between them must be zero. Similarly, streams which are desynchronized with respect to each other must have a non-zero value for this quantity. Note that the value of the degree of asynchrony, $d_t^{i,j}$, may be negative although the Sub-word State_t^i variables are always non-negative. In order to prevent the model from hypothesizing implausible articulatory alignments, we constrain the maximum allowable asynchrony between feature streams by

imposing an upper-bound, M, on the absolute value of degree of asynchrony between any pair of articulatory feature streams:

$$-M \leq d_t^{i,j} \leq M \qquad \text{for all } 1 \leq i, j \leq K \text{ and for all } 1 \leq t \leq T \tag{3.3}$$

where, T is the length of the utterance. In the DBN-based model presented in Figure 3.2, these constraints are imposed using the FtrAsyncConfig, AsyncConfig and CheckAsyncConfig variables.

The variable FtrAsyncConfig$_t$ is intended to represent the current configuration of asynchrony amongst all the variables in the system. For concreteness, this shall be computed by specifying the degree of asynchrony of all the streams with respect to one reference streams, say stream 1. Thus, if we compute the degree of asynchrony with respect to the first feature stream, FtrAsyncConfig$_t$ represents the unique configuration $(d_t^{2,1}, d_t^{3,1}, \cdots, d_t^{K,1})$. For example, FtrAsyncConfig$_t$ $= (1, 0, \cdots, 0)$, would indicate that the second articulatory stream is one state ahead of the first stream, while the remaining streams are synchronized with the first. From Equation 3.3, each of the degrees of asynchrony have cardinality $2M + 1$, and thus FtrAsyncConfig has cardinality of at most $(2M + 1)^{K-1}$. However, not all of these configurations will be valid according to Equation 3.3. For example, a configuration where $d_t^{2,1} = M$ and $d_t^{3,1} = -M$ is invalid, since in this case stream 2 is $2M$ states ahead of stream 3, (i.e., $d_t^{2,3} = 2M$), and thus is inadmissible under the constraint in Equation 3.3. AsyncConfig$_t$ is a vector-valued variable with no parents that can take on any value corresponding to an allowable asynchrony configuration. Finally, the variable CheckAsyncConfig$_t$ is a 'dummy' variable, that has probability 1 if and only if FtrAsyncConfig$_t$ and AsyncConfig$_t$ have the same value.[21] Thus, the set of asynchrony

[21] Dummy variables are a standard mechanism for representing what are essentially symmetric constraints in a directed model. The dummy variables can be represented in the DBN as variables with cardinality 2, and

variables are designed to ensure that each frame of the model contributes a value corresponding to the prior on the asynchrony configuration towards the overall probability of the assignments for the full sequence, while simultaneously ensuring that inadmissible combinations are assigned zero-probability in the model. Thus, the distribution of AsyncConfig, which is learned during DBN training, represents the probability of each asynchrony configuration.[22]

3.5 CRF-based Model for Articulatory Feature Alignment

A proposed model for CRF-based articulatory feature alignment that incorporates a number of features drawn from the DBN-model is shown in Figure 3.5 in the form of a factor graph [Kschischang et al., 2001].[23] For reference, we also draw the corresponding undirected graphical model in Figure 3.6. The factor graph, or equivalently the structure of cliques in the undirected graphical model, make explicit the conditional independence assumptions in the corresponding CRF. The factor nodes in the graph (represented as red and blue squares) in Figure 3.5 represent (non-negative) functions over the set of variable nodes (represented by circles) that are connected to it in the graph. As before, we denote the sequence of observations by $\overline{\mathbf{x}}$, with the particular observation at time t denoted by $\mathbf{x}_t$

can take on either value 0 or 1. The conditional probability table of the dummy variable conditioned on its parents is then expressed in such a manner that only admissible configuration of values of its parents would result in it taking on the value 1 (say) with non-zero probability. Since the dummy variables are all observed variables with value 1, these variables have the effect of only allowing admissible configurations of variables to receive non-zero probability in the full 'unrolled' sequence (over time) of variables. As we shall see in section 3.5 such variables are unnecessary in the undirected CRF model.

[22]Incorporating both the deterministic FtrAsyncConfig as well as the non-deterministic AsyncConfig variables in the model allows the asynchrony distribution to be learned during the DBN training via the Expectation-Maximization algorithm [Dempster et al., 1977]. See [Livescu and Glass, 2004b] for details.

[23]In the interest of brevity, we do not provide a review of conditional random fields. Interested readers are referred to the many references on this topic [Lafferty et al., 2001; Sutton and McCallum, 2012; Fosler-Lussier et al., 2013]

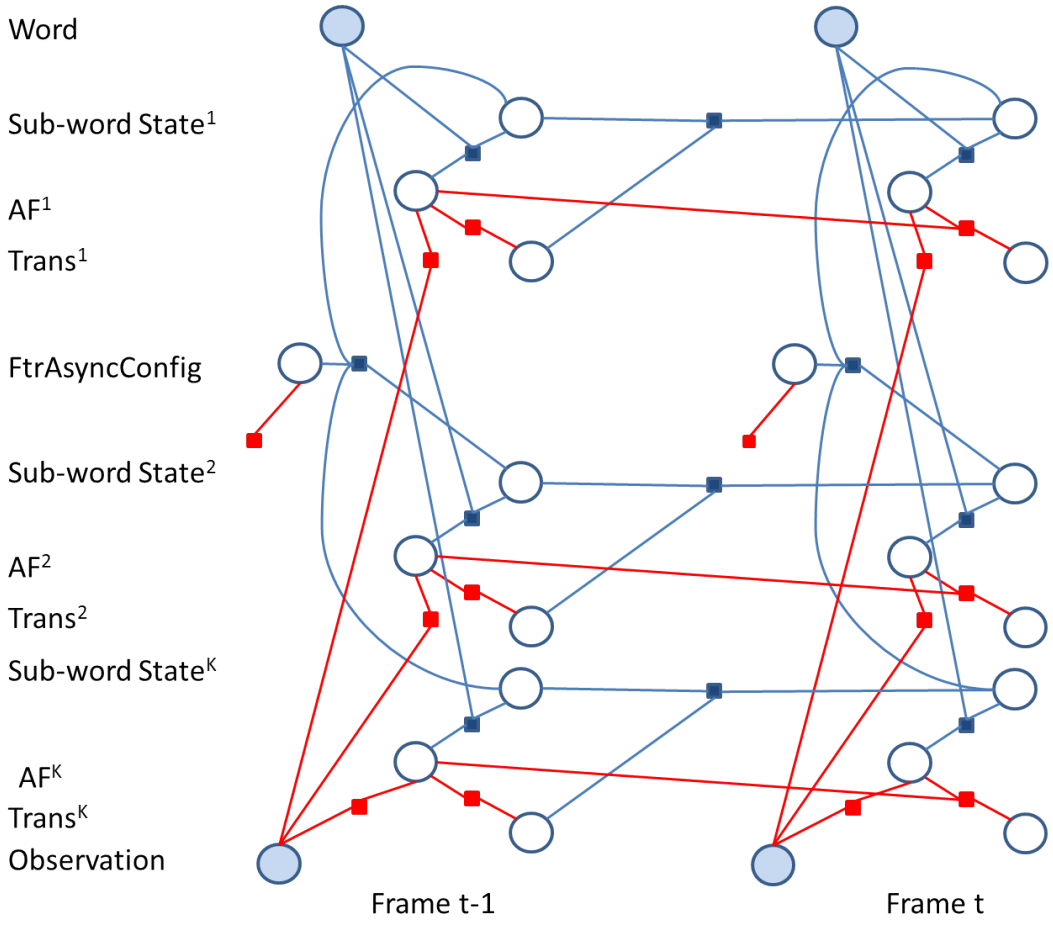

Figure 3.5: Factor graph representing the proposed CRF model for articulatory feature alignment. Corresponding undirected graphical model appears in figure 3.6. The shaded nodes represent variables that we condition on. The red and blue square nodes represent factors: non-negative functions defined over the configurations of the set of variables connected to it.

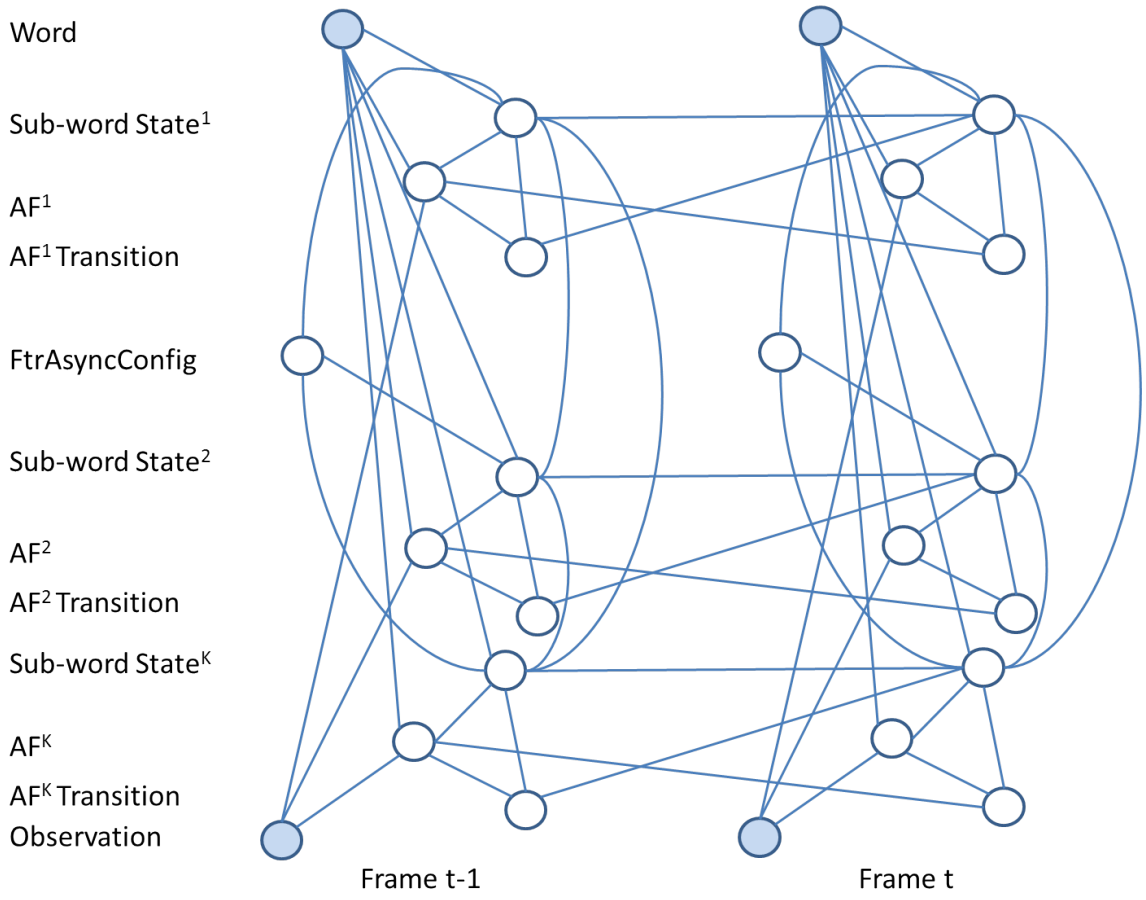

Figure 3.6: Undirected graphical model representing the proposed CRF model for articulatory feature alignment. Corresponding factor graph appears in figure 3.5.

and the word that the utterance corresponds to by $\overline{v}$. The sequence of all other variables in the graph is represented by $\overline{\mathbf{y}}$, with $\mathbf{y}_t$ being the set of all of these variables at time t.

Each factor node can be associated with a particular time t. We assume that a factor can be indexed by the set of variable nodes, $c \in \mathcal{C}$, that it is connected to in the graph. Let $\mathbf{y}_t^c$ represent the set of variables associated with this set of variables at a particular time t. Note that the set of variables c form a clique in the undirected graphical model in figure 3.6 and they can, thus, be thought of as 'clique templates' in the dynamic conditional random field model of Sutton et al. [Sutton et al., 2004]. The probability distribution over the sequence of variables $\overline{\mathbf{y}}$, conditioned on the observations $\overline{\mathbf{x}}$ and the word $\overline{v}$ in the factor graph can then be expressed as a normalized product of potentials corresponding to the factors in the graph, with the only constraint that the potentials $\phi_c(\mathbf{y}_t^c, \overline{\mathbf{x}}, \overline{v}, t)$ be non-negative,

$$P(\overline{\mathbf{y}}|\overline{\mathbf{x}}, \overline{v}) = \frac{1}{Z(\overline{\mathbf{x}}, \overline{v})} \prod_t \prod_{c \in \mathcal{C}} \phi_c(\mathbf{y}_t^c, \overline{\mathbf{x}}, \overline{v}, t) \tag{3.4}$$

where, $Z(\overline{\mathbf{x}}, \overline{v})$ is a normalization term that ensures that we have a valid probability distribution.

Deterministic vs. Trainable factors

In our model, we distinguish between factors that are associated with learnable parameters of the model, denoted in red in Figure 3.5, from those that enforce deterministic constraints, denoted in blue in Figure 3.5. We shall elaborate on the differences between the two types of factors shortly, but for now it suffices to say that factors with trainable parameters are associated with a vector of pre-defined 'feature functions', $\mathbf{f}^c(\mathbf{y}_t^c, \overline{\mathbf{x}}, \overline{v}, t) = [\cdots, f_i^c(\mathbf{y}_t^c, \overline{v}, \overline{\mathbf{x}}, t), \cdots]^T$ where $1 \leq i \leq N_c$, for some integer N_c. We then model the potential associated with this factor as,

$$\phi_c(\mathbf{y}_t^c, \overline{\mathbf{x}}, \overline{v}, t) = e^{\mathbf{w}_c \cdot \mathbf{f}^c(\mathbf{y}_t^c, \overline{\mathbf{x}}, \overline{v}, t)} \tag{3.5}$$

where, $\mathbf{w}_c$ is a vector of weights to be learned during training. In Equation 3.5, we have implicitly assumed that weights are tied across repeating clique-templates across various frames at different times t as in the dynamic CRF model [Sutton et al., 2004].

In our model the trainable factors are associated with configurations of feature asynchrony, individual articulatory features (the "acoustic model"), and articulatory feature transitions (the "transition model"). The feature functions associated with each articulatory feature variable AF_t^i are constructed by first computing a set of statistics $g_{l,m}(\mathbf{x}_t)$ from the acoustics (e.g. $g_{l,m}(\mathbf{x}_t)$ could be the l^{th} output of a particular multilayer perceptron (MLP) indexed by m, as in Section 3.6). These statistics are then used to construct individual components in the vector of feature functions associated with the articulatory feature variable (AF^i),

$$f_{i,j,l,m}(\text{AF}_t^i, \overline{\mathbf{x}}, t) = g_{l,m}(\mathbf{x}_t)\delta(\text{AF}_t^i = a_j^i) \tag{3.6}$$

where a_j^i is one value that AF^i can be assigned and $\delta(z = z') = 1$ if $z = z'$ and 0 otherwise. The feature functions associated with the feature asynchrony configuration (FtrAsyncConfig) and articulatory feature transitions ($\text{AF}^i, \text{Trans}^i$) are

$$f_r(\text{FtrAsyncConfig}_t, t) = \delta(\text{FtrAsyncConfig}_t = r) \tag{3.7}$$

$$f_{i,j,v}(\text{AF}_{t-1}^i, \text{AF}_t^i, \text{Trans}_t^i, t) = \delta(\text{AF}_{t-1}^i = a_j^i)\delta(\text{AF}_t^i = a_{j'}^i) \tag{3.8}$$

where r is an asynchrony configuration vector as defined in Section 3.4 and $a_j^i, a_{j'}^i$ are possible values that can be assigned to AF^i.

The deterministic factors in our work shall be binary (zero-one) functions, the only purpose of which shall be to ensure that certain 'invalid' sequences of assignments to the variables $\overline{\mathbf{y}}$ are assigned zero probability by the model. Specifically, for the model in Figure 3.5, since the sub-word state indices in the model are meant to represent the index in

the pronunciation of the word, any valid assignment to the sub-word indices must ensure that:

- sub-word state indices increment by at most 1 between adjacent frames,

$$0 \leq \text{Sub-word State}_{t+1}^{i} - \text{Sub-word State}_{t}^{i} \leq 1 \tag{3.9}$$

- at the first time-frame, the articulators be in the first unit of the pronunciation,

$$\text{Sub-word State}_{1}^{i} = 1 \tag{3.10}$$

- the last frame corresponds to the last unit in the pronunciation of the word where $\text{pron}(w)$ is the pronunciation of the word in terms of articulatory features,

$$\text{Sub-word State}_{T}^{i} = |\overline{v}| \tag{3.11}$$

- and finally that the articulatory feature value AF^{i} is given by the value specified in the words pronunciation for that articulatory feature for a particular value of the corresponding sub-word state,

$$\text{AF}_{t}^{i} = \sigma_{k}^{i} \quad \text{if Sub-word State}_{t}^{i} = k \tag{3.12}$$

For example, the condition expressed in equation 3.10 would be expressed using a deterministic factor as,[24]

$$\phi(\text{Sub-word State}_{1}^{i}, \mathbf{x}, w, 1) = \begin{cases} 1 & \text{Sub-word State}_{1}^{i} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

[24]In this and following equations, we abuse notation slightly by not *explicitly* specifying the clique c that the potentials are associated with, as we did in previous equations since it is clear from context what the associated cliques are.

3.5.1 Simplifying the Model

In pilot experiments, we implemented the CRF shown in Figure 3.5 using the GRMM toolkit [Sutton, 2006]. However, we found that performing exact inference in this model using the toolkit was prohibitively slow, and the use of approximate inference algorithms resulted in poor performance. A likely hypothesis which might explain the slowness of exact inference in the toolkit is that it is due to the fact that the toolkit does not automatically exploit the sparsity that results from deterministic constraints in our model. In this section, we describe how we take advantage of this sparsity to allow us to do fast exact inference in our CRF.

Although the original CRF model shown in Figure 3.5 appears complicated, the deterministic constraints in the model allow for efficient *exact* inference. The first observation that we make is that we can eliminate a number of 'deterministic' variables (variables that are deterministically determined by configurations of other variables in the model) if we allow for the creation of more general feature functions.

As a specific example, consider the feature function in Equation 3.6. Since the word that the utterance corresponds to is known and fixed, we we can re-write the above feature function to depend directly on the sub-word state corresponding to the articulatory feature stream. Exploiting the fact that AF_t^i is deterministically determined given $\text{Sub-word State}_t^i$ and $\overline{v}$, we can restate the feature functions in Equation 3.6-3.8 as,

$$f_i, j, l, m(\text{Subword State}_t^i, \overline{\mathbf{x}}, \overline{v}, t) = g_{l.m}(\mathbf{x}_t)\delta(\sigma^i_{\text{Sub-wordState}_t^i} = a_j^i) \tag{3.14}$$

$$f_r(\text{SubwordState}_t^1, \text{SubwordState}_t^2, \cdots, \text{SubwordState}_t^K, t) = \delta((d_t^{2,1}, d_t^{2,1}, \cdots, d_t^{K,1}) = r) \tag{3.15}$$

$$f_{i,j,j'}(\text{SubwordState}_{t-1}^i = s', \text{SubwordState}_t^1 = s, \overline{v}, t) = \delta(\sigma^i_{s'} = a^i_{j'})\delta(\sigma^i_s = a^i_j) \tag{3.16}$$

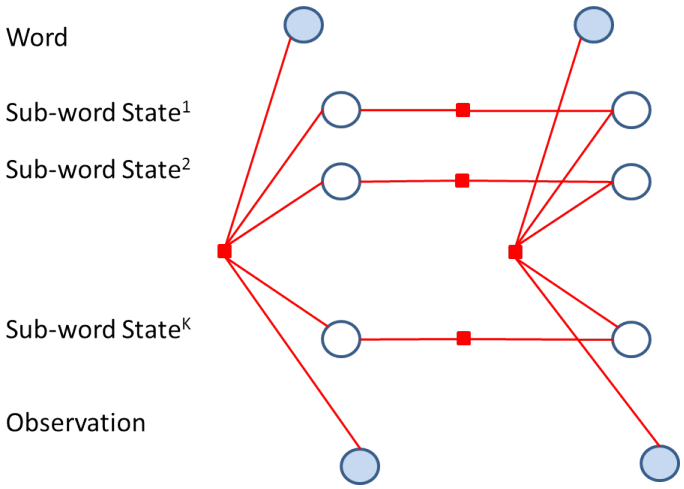

Figure 3.7: Factor graph representation of simplified model with 'deterministic variables' removed.

where, $d_t^{i,j}$ is as defined in Equation 3.2, $a_j^i, a_{j'}^i$ represent particular values corresponding to the i-th articulatory feature stream. and r represents a relative asynchrony configuration. By eliminating such 'deterministic' variables, observe that an equivalent model to the CRF in Figure 3.5 can be obtained by representing only the sub-word state variables, corresponding to the K feature streams as shown in Figure 3.7. In essence, the model has been simplified by reducing the number of variables, with additional complexity in the feature functions. Note however, that some of the deterministic constraints in the original model, in particular, Equations 3.9–3.11 must still be retained in the simplified model. Finally, we note that we can obtain an equivalent model by collapsing the K sub-word state variables into a single variable, $\text{SubwordConfiguration}_t$ whose domain is the cross-product of the individual SubwordState_t variables: $\text{SubwordConfiguration}_t = (\text{SubwordState}_t^1, \cdots, \text{SubwordState}_t^K)$. The resulting model is depicted in Figure 3.8. We stress here that this process of collapsing the articulatory variables into a single variable is purely for convenience; the transformed model is exactly equivalent to the original factored

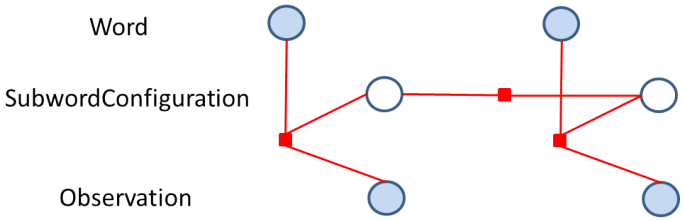

Figure 3.8: Factor graph representation of final simplified model after sub-word state variables have been collapsed to obtain a linear chain.

CRF model in Figure 3.5, since we ensure that the feature functions (both deterministic and trainable) constructed on the Sub-wordConfiguration variable in the resulting model are exactly the same as those in the original model. In the model that appears in Figure 3.8, the factor corresponding to the state transition includes all potentials involving only these variables (Equations 3.16 and 3.9), the second factor includes the local acoustic features (Equation 3.14), the asynchrony features (Equation 3.15) and the deterministic potentials (Equations 3.10-3.12) encoding the dictionary.

3.5.2 Efficient Exact Inference

In previous sections, we described a process by which the original CRF model of Figure 3.5 can be converted into an equivalent linear chain model by choosing a particular sequence of variable and factor clustering operations [Kschischang et al., 2001] or equivalently a particular triangulation of the original undirected graphical model. In this section, we describe how the deterministic factors in the model allow for *efficient* and *exact* inference in the resulting *equivalent* model.

We begin with some additional notation: denote by $|\overline{v}|_{\max}$ the maximum length of the pronunciation of any word,

$$|\overline{v}|_{\max} = \operatorname*{argmax}_{\overline{v}} |\overline{v}| \tag{3.17}$$

Input: word, $\overline{v}$; acoustics, $\overline{\mathbf{x}}$; number of articulatory streams, K; number of units in pronunciation, $|\overline{v}|$

Initialize: $\text{Candidates}(1) = \{(1, 1, \cdots, 1)\}$

Determine possible configurations:

For $t = 2$ to T

$\text{Candidates}(t) = \text{next}(\text{Candidates}(t-1), \overline{v})$

Initialize boundary conditions:

For $t = 1$ to T

For $s \in \text{Candidates}(T)$

$\alpha_t(s) = 0$;

$\beta_t(s) = 0$;

Compute alpha recursively:

$\alpha_1((1, 1, \cdots, 1)) = \exp^{\mathbf{w} \cdot \mathbf{f}((1,1,\cdots,1), \overline{\mathbf{x}}, \overline{v}, t)}$

For $t = 2$ to T

For $s \in \text{Candidates}(t)$

$\alpha_t(s) = \sum_{s' \in \text{prev}(s, \overline{v})} \alpha_{t-1}(s') e^{\mathbf{w} \cdot \mathbf{f}(s', s, \overline{\mathbf{x}}, \overline{v}, t)}$

Compute beta recursively:

$\beta_T((|\overline{v}|, |\overline{v}|, \cdots, |\overline{v}|)) = 1$

For $t = T - 1$ down to 1

For $s \in \text{Candidates}(t)$

$\beta_t(s) = \sum_{s' \in \text{next}(s, \overline{v})} \beta_{t+1}(s') e^{\mathbf{w} \cdot \mathbf{f}(s, s', \overline{\mathbf{x}}, \overline{v}, t)}$

Return required marginals:

$$P(s_{t-1} = s, s_t = s' | \overline{\mathbf{x}}, \overline{v}) = \frac{1}{\alpha_T((|\overline{v}|, |\overline{v}|, \cdots, |\overline{v}|))} \alpha_{t-1}(s) e^{\mathbf{w} \cdot \mathbf{f}(s, s', \overline{\mathbf{x}}, \overline{v}, t)} \beta_t(s')$$

Figure 3.9: Sum-product algorithm for computing marginal distributions for the model that appears in Figure 3.8.

We use the notation $\mathbb{N}_k = \{1, 2, \cdots, k\}$, to denote the set of natural numbers less than or equal to k. With this notation, the domain for the variable Sub-wordConfiguration$_t$ is the set $\mathbb{N}_{|\overline{v}|_{\max}} \times \mathbb{N}_{|\overline{v}|_{\max}} \times \cdots \times \mathbb{N}_{|\overline{v}|_{\max}} = \mathbb{N}^K_{|\overline{v}|_{\max}}$. We define the function next$(s, \overline{v})$ for the $s \in$ Sub-wordConfiguration$_t$, that encodes the set of values that the variable can take in the next time-step. Given $s \in \mathbb{N}^K_{|\overline{v}|_{\max}}$ and a word $\overline{v}$, we define,

$$\text{next}(s, \overline{v}) = \Big\{ x \in \mathbb{N}^K_{|\overline{v}|_{\max}} \quad | \quad \text{Sub-wordConfiguration}_t = s \text{ and Sub-wordConfiguration}_{t+1} = x \text{ satisfy the deterministic constraints in Equations 3.9–3.11} \Big\} \tag{3.18}$$

For example, if we have three articulatory feature streams ($K = 3$), the maximum allowed asynchrony between adjacent feature streams is one state ($M = 1$), and if the word $\overline{v}$ has three units in its pronunciation ($|\overline{v}| = 3$), we have,

$$\text{next}((1,1,1), \overline{v}) = \{(1,1,1), (2,1,1), (1,2,1), (1,1,2), (2,2,1), (2,1,2), (1,2,2), (2,2,2)\} \tag{3.19}$$

$$\text{next}((1,2,1), \overline{v}) = \{(1,2,1), (1,2,2), (2,2,1), (2,2,2)\} \tag{3.20}$$

$$\text{next}((2,3,3), \overline{v}) = \{(2,3,3), (3,3,3)\} \tag{3.21}$$

$$\text{next}((3,3,3), \overline{v}) = \{(3,3,3)\} \tag{3.22}$$

$$\text{next}((4,3,3), \overline{v}) = \{\} \tag{3.23}$$

In Equation 3.19 any of the sub-word states may be incremented without violating any constraint, unlike in the remaining examples. Additionally, since we assumed that $\overline{v}$ has exactly three units in its pronunciation, the configuration $(4, 3, 3)$ would be invalid for this word. This also explains why next$((3, 3, 3), \overline{v})$ in example 3.22 contains only the element $(3, 3, 3)$. The function $\text{prev}(s, \overline{v})$ is defined analogously to encode the set of sub-word configurations that can precede any particular configuration s given a word $\overline{v}$. Finally, we extend the notation by allowing the $\text{next}(s, \overline{v})$ and $\text{prev}(s, \overline{v})$ functions to be defined on

sets of configurations as well,

$$\text{next}(\{s_1, s_2, \cdots, s_k\}, \overline{v}) = \bigcup_{i=1}^{k} \text{next}(s_i, \overline{v}) \tag{3.24}$$

$$\text{prev}(\{s_1, s_2, \cdots, s_k\}, \overline{v}) = \bigcup_{i=1}^{k} \text{prev}(s_i, \overline{v}) \tag{3.25}$$

With these definitions, we can perform inference on the factor graph to obtain the marginal distributions over Sub-wordConfiguration$_t$ at time t using the standard sum-product algorithm [Kschischang et al., 2001], which in this case is equivalent to the standard alpha-beta recursions for linear-chain CRFs [Lafferty et al., 2001]. For completeness, we state the alpha-beta recursions for the problem in Figure 3.9. Note that the deterministic constraints have been explicitly captured in the algorithm since we explicitly ensure that the quantities are only computed over pairs of configurations s, s' where $s' \in \text{next}(s, \overline{v})$ or $s' \in \text{prev}(s, \overline{v})$ as the case may be. In other words, the deterministic constraints have been implicitly encoded in the $\text{next}(\cdot, \cdot)$ and $\text{prev}(\cdot, \cdot)$ functions. Apart from this restriction and the explicit construction of the candidate sets, the algorithm is essentially the same as the standard recursion employed in linear-chain CRFs.

3.5.3 Analysis of Complexity of Computing Marginal Distributions using the Algorithm in Figure 3.9

The equivalent linear-chain CRF in Figure 3.8, in which all of the sub-word state variables have been collapsed together, has a large state space corresponding to the Cartesian product of the domains of the sub-word state variables. Since, $|\overline{v}|_{\text{max}}$ is the largest allowable value of the sub-word state, a naive analysis shows that there are $|\overline{v}|_{\text{max}}^{K}$ states in the label space. Since inference in linear chain CRFs is known to be quadratic in the labels, we have an overall complexity of $O(|\overline{v}|_{\text{max}}^{2K} T)$ which would seem to be prohibitive. The complexity of the algorithm in Figure 3.9 is however much less as result

of the deterministic constraints in the model. Since we assume that the maximum allowable asynchrony between any pair of feature streams is M, the possible valid assignments to the configuration of sub-word states must be less than or equal to $|\overline{v}|_{\max} (2M+1)^{K-1}$ for the K streams. To see this, notice that the sub-word state corresponding to the first articulatory feature stream can take on any of the $|\overline{v}|_{\max}$ values in $\mathbb{N}_{|\overline{v}|_{\max}}$. However, the next stream can be desynchronized from it by at most M units or else be completely synchronized. Similar observations hold for the remaining streams. Thus, in the algorithm, $|\operatorname{Candidates}(t)| \leq |\operatorname{Candidates}(T)| \leq |\overline{v}|_{\max} (2M+1)^{K-1}$. Since, in our experiments we set $M = 1$ and $K = 3$, the total number of states that needs to be considered is only *linear in the number of units in the pronunciation of the word*.

Finally, note that the during the computation of the alpha-beta recursions in the algorithm, we are only required to sum over the states $s' \in \operatorname{next}(s, w)$ or $s' \in \operatorname{prev}(s, w)$. However, since each sub-word state variable can either increment or else remain the same between adjacent time-steps, $|\operatorname{next}(s, w)| \leq 2^K$. A similar analysis shows that $|\operatorname{prev}(s, w)| \leq 2^K$. Thus, the overall complexity of the algorithm in Figure 3.9 is $O(T 2^K |\overline{v}|_{\max} (2M+1)^{K-1})$. Note that collapsing all of the variables at each frame in Figure 3.5 directly and applying standard inference algorithms for the corresponding linear chain model would incur quadratic complexity in the size of the cross-product of the cardinalities of the variables at each frame $O(T|\overline{v}|^2 K 2^2 K (2M+1)^{2(K-1)} \left|\mathrm{AF}^1\right|^2 \left|\mathrm{AF}^2\right|^2 \cdots \left|\mathrm{AF}^K\right|^2)$. Exploiting task-specific constraints allows us to reduce training time by many orders of magnitude: In our experiments, each pass through the training data takes less than a minute, whereas using an off-the-shelf inference engine [Sutton, 2006] took approximately a day.

Set	Number of words	Number of frames
Train	2941	89748
Development	165	5365
Test	236	7037

Table 3.1: Statistics for train, development and test data for the subset of STP [Greenberg et al., 1996] used in our experiments.

3.6 Experiments

In order to determine the effectiveness of the proposed CRF-based AF-alignment model, we conducted experiments on a subset of the Switchboard Transcription Project (STP) data [Greenberg et al., 1996]. The STP data contains a subset of conversational telephone speech recorded as part of the Switchboard corpus [Godfrey et al., 1992], that was transcribed manually at the phonetic level using a phone-set derived from TIMIT [Garofolo et al., 1993]. The transcriptions also contain additional diacritical information that indicate, for example, nasalization, frication (of a normally unfricated segment), etc.

Dataset Selection

The data used for this experiment was identical to the data used in lexical access experiments by Livescu [Livescu, 2005]: we extract data for all words from the "train-ws96-i" subset if they belong to the set of the 3500 most likely words in the Switchboard corpus, after excluding partial-words and filled-pauses etc. This data was then divided into a training set (data from sets 24–49), a held-out development set (set 20) used to tune parameters and a test set (sets 21–22) that we report results on.

For each of the train, development and test sets, we excise the speech utterances corresponding to individual words based on the information in the corresponding word transcripts. The words so excised from the training set form our set of training examples; words

from the development and test sets are used to evaluate the performance of the baseline and proposed systems. Statistics of the data used in the experiment appear in Table 3.1.

Obtaining Ground Truth Articulatory Features

For each word in the train, development and test sets, we extract phone transcriptions for the word by aligning the word and phone transcripts. Since these do not align exactly, we consider a phone to be part of a word's transcription if the phone boundary is at least 10ms within the boundary of the corresponding word. We split stops, affricates and diphthongs into two segments, representing the initial and final portion of the phone. The first two-third portion of the original phones duration is assigned to the first segment, while the remaining one-third is assigned to the second. After time-aligned phone labels have been obtained for each word in this manner, we strip away all diacritic information other than nasalization and map the phone to obtain the corresponding articulatory feature labels using the deterministic mapping outlined in [see Livescu, 2005, Appendix B], which serve as the ground truth.[25]

Details of the Feature-based Pronunciation Model

In all our experiments, we assume that the tongue tip and tongue body features (T), the lip features (L) and the combination of glottis and velum (G) are each completely synchronized, thus resulting in a model with three effective feature streams ($K = 3$). The maximum allowable asynchrony between any pair of streams is set to one state ($M = 1$). Considering these constraints the number of distinct L, T and G labels was 8, 25 and 4 respectively. Additional details of the features and the phone-to-feature mappings can be found in [see Livescu, 2005, Appendix B].

[25]This is, of course, not ideal since the phone labels are not necessarily an accurate representation of the articulatory configurations [Saraçlar and Khudanpur, 2004].

Acoustic Parameterization: Training Multilayer Perceptrons

We parameterize the acoustics by computing 12th-order speaker-normalized PLPs with energy, deltas and double-deltas to obtain a 39-dimensional input representation ($\mathcal{X} \subseteq \mathbb{R}^{39}$). We train three multilayer perceptrons (MLPs) to predict each of the L, T, G features using the phone-derived AF labels as well as an MLP to predict the underlying phone (corresponding to the STP [Greenberg et al., 1996] phoneset). The feature vectors for a given frame are concatenated with the four preceding and succeeding frames to obtain a 351-dimensional input representation to the MLPs. The MLPs are single hidden layer feed-forward networks with a sigmoid activation function for hidden layer nodes and a softmax activation function for the output layer. The MLPs are trained using the Quicknet toolkit [Johnson et al., 2004] to optimize a cross-entropy-based criterion with the number of hidden layer nodes determined by tuning MLP frame-level accuracy on the development set.

When constructing CRF feature functions according to Equations 3.14-3.16, we consider three statistics derived from the MLPs: (a.) *posteriors* (CRF-Post), the softmax outputs from the MLPs, (b.) *log posteriors* (CRF-LogPost), obtained by computing the logarithm of the softmax outputs from the MLPs, and (c.) *linear outputs* (CRF-Lin), obtained by removing the final softmax output layer from the MLPs.[26]

We compare the performance of the CRF-based systems against three baseline DBN systems. The first axis along which the DBN systems vary is in the representation of the

[26]If u_i represents the weighted sum of hidden layer activations of the MLP corresponding to class $y = i$, then the *softmax* output for class i – the posterior output, z_i – is computed as $z_i = \frac{e^{u_i}}{\sum_j e^{u_j}}$, which represents the posterior probability, $P(y = i|\mathbf{x})$, of the ith class conditioned on the acoustics $\mathbf{x}$ at that frame. The log posterior and linear outputs corresponding to the ith class are respectively $u_i - \log \sum_j e^{u_j}$ and u_i respectively. Thus, the linear outputs and the log posteriors differ from each other by an additive factor that is dependent on the acoustics $\mathbf{x}$ at that frame.

variable $\overline{\mathbf{x}}$ in Figure 3.2: (a.) a system that represents the variable $\overline{\mathbf{x}}$ in the model as PLP coefficients, and (b.) a system that represents the variable $\overline{\mathbf{x}}$ as the linear outputs of the MLPs projected onto the top 39 principal components after Principal Components Analysis to obtain a "tandem" representation [Hermansky et al., 2000] (DBN-Tandem). We further consider two variants of the PLP-based DBN systems: (a.) a system that does not allow for any asynchrony amongst articulatory feature streams (DBN-PLP-noasync) and is similar to a phone-based model, and (b.) a system that allows for up to one state of asynchrony among pairs of feature streams ($M = 1$) (DBN-PLP-async). The DBN-tandem and CRF systems always allow for up to one frame of asynchrony ($M = 1$).

Baseline DBN and CRF systems

The baseline DBN systems, described in Section 3.4, are implemented using the Graphical Models Toolkit (GMTK) [Bilmes and Zweig, 2002]. The output distributions of the acoustics – $p(\overline{\mathbf{x}}|L, T, G)$ – are modeled as mixtures of Gaussians, with the number of Gaussians for each L, T, G configuration determined using the splitting-vanishing procedure [Bilmes and Zweig, 2002]. The optimal number of Gaussians for each (L, T, G) configuration was determined based on frame-level articulatory feature classification accuracy on the development set.

Since training the CRF-based systems is a supervised learning problem, we require time-aligned AF labels[27] in order to train the CRF-based systems. For this purpose, we use the fully trained DBN-PLP-async system to obtain AF transcriptions for the training set

[27]Since we require AF segmentations that satisfy the assumptions in the pronunciation model, e.g., that there is no articulatory feature substitution and that the maximum amount of allowed asynchrony is one state ($M = 1$), the AF labels obtained by deterministically mapping the phone labels to AF targets cannot be used directly for CRF training, although they can be used for training the MLPs.

Set	System Type	L Err. Rate (%)	T Err. Rate (%)	G Err. Rate (%)	Joint Err. Rate (%)
Train	DBN-PLP-async	11.2	34.7	15.9	20.6
Dev	DBN-PLP-async	10.8	31.0	16.6	41.6
	DBN-PLP-noasync	11.2	31.8	15.8	38.2
	DBN-Tandem	11.4	30.6	18.0	41.1
	CRF-Post	10.6	$27.0^{*\dagger\ddagger}$	$12.4^{*\dagger\ddagger}$	$35.2^{*\dagger\ddagger}$
	CRF-LogPost	$9.4^{*\dagger\ddagger}$	$26.6^{*\dagger\ddagger}$	$13.1^{*\dagger\ddagger}$	$34.4^{*\dagger\ddagger}$
	CRF-Lin	$9.8^{*\dagger\ddagger}$	$27.1^{*\dagger\ddagger}$	$13.4^{*\dagger\ddagger}$	$35.0^{*\dagger\ddagger}$
Test	DBN-PLP-async	9.6	35.2	16.8	44.0
	DBN-PLP-noasync	9.3	35.2	16.0	40.6
	DBN-Tandem	9.9	35.4	17.7	43.7
	CRF-Post	10.6	$33.3^{*\dagger\ddagger}$	$14.9^{*\dagger\ddagger}$	$40.5^{*\ddagger}$
	CRF-LogPost	9.6	$33.4^{*\dagger\ddagger}$	$14.8^{*\dagger\ddagger}$	$39.7^{*\ddagger}$
	CRF-Lin	9.2	$32.8^{*\dagger\ddagger}$	$14.4^{*\dagger\ddagger}$	$40.0^{*\ddagger}$

Table 3.2: Frame-level error rates for forced-transcription experiments obtained on the various sets using the DBN and CRF systems. (*,†, ‡) indicate statistically significant improvements ($p \leq 0.05$) over the DBN-PLP-async, DBN-PLP-noasync, and DBN-Tandem systems respectively using a one-tailed Z-test.

given the identity of the word.[28] These are then used as training labels for the CRF-based system.

3.7 Results

We present frame-level error rates measured against the phone-derived articulatory feature labels in Table 3.2 for each of the articulatory feature streams as well as for the joint configuration of all the articulators. Since the CRFs are trained on labels obtained by force-aligning the training set using DBN-PLP-async system, we also present error rates on the training data. As can be seen from the table, the error rates on the training data, and hence our training labels, are fairly noisy.

[28]These decoded AF target labels satisfy the constraints of the model.

On the development set, both CRF systems employing log-posterior as well as the linear features outperform all DBN systems with relative error rate reductions from between 7.8%–31.0% across the various systems and feature categories. On the test set, however, there are no significant differences in terms of classification of the lip (L) feature but we do see gains in classification of the T and G features, and a small improvement in joint classification accuracy. In these cases, the CRF-based systems result in relative error rate reductions of between 5.0% and 18.6%.

3.8 Discussion

The results presented in Table 3.2 are encouraging for two reasons. Firstly, in these pilot experiments we explored a very limited set of feature functions in the CRF. In previous work [Morris, 2010] CRFs have been shown to be effective combiners of MLP-based feature detectors. We can therefore hope that similar improvements may be possible in this domain by incorporating additional feature classifiers as well. Additionally, it might be beneficial to incorporate additional feature functions in order to obtain a better model of articulatory asynchrony. For example, a more detailed model of articulatory asynchrony might include information about position of the unit within the word, unigram language model probability of the word, measures of speaking rate, and other factors that are known to be correlated with pronunciation variation. Secondly, recall that the training labels for the CRF are derived using the baseline DBN system. The error rate of the DBN on the training data is quite high, and it is encouraging to see that the CRF can still outperform the DBN system on the development and test set. In fact, gains are observed in the CRF-based systems in spite of the fact that the asynchronous DBN system (DBN-PLP-async) does not outperform the DBN system that does not allow any asynchrony (DBN-PLP-noasync).

Two possible reasons for the high error rates in the DBN are: (a.) lack of sufficient training data or (b.) the model that we have considered does not allow for feature substitution and is hence limited in its ability to model pronunciation variation. In principle, a model that incorporates feature substitution (which would model reduction in gestural magnitudes, cf. 2.2), might result in improved performance. However, adding in a model for AF-substitution would require significantly more complexity in decoding.

3.9 Summary

In this chapter, we presented a conditional random field-based model for articulatory feature forced-transcription. In experimental results conducted on the Switchboard Transcription Project data [Greenberg et al., 1996], the proposed CRF models were found to improve performance significantly over dynamic Bayesian networks presented previously [Livescu, 2005] for this task. In experimental evaluations, the proposed techniques resulted in improvements of between 5.0%–18.6% over the baselines in predicting tongue and glottis/velum features.

In Chapters 5 and 6, we apply the models of pronunciation developed in this chapter for the task of discriminative spoken term detection (STD) in conversational speech settings. Before we incorporate the AF-based pronunciation model within the STD system, we first study the effectiveness of the proposed STD systems in the context of phone-based pronunciation modeling. Thus, the experiments in the following chapter serve as a means of validating the proposed techniques in the 'simpler' setting of phone-based pronunciation modeling before we tackle AF-based pronunciation modeling for STD in subsequent chapters.

CHAPTER 4: DISCRIMINATIVE SPOKEN TERM DETECTION IN LOW-RESOURCE SETTINGS

In this chapter, we study the problem of spoken term detection (STD)[29] – the problem of detecting whether or not a set of pre-defined terms (keywords) are present in a set of speech utterances (along with their locations) – in the context of low-resource settings where labeled training data are limited.[30] As was mentioned in the previous chapter, our final goal is to incorporate the articulatory feature-based (AF-based) pronunciation model within an STD system. The experiments presented in this chapter are a a step in this direction and are aimed at examining some aspects of the proposed discriminative STD approach. However, instead of directly applying them to AF-based pronunciation models, we begin by examining phone-based pronunciation models. This allows us to examine aspects of the proposed model in the 'simpler' setting of phone-based pronunciation models. We return to the problem of STD using AF-based pronunciation models in Chapters 5 and 6.

In this chapter, we present one of the main contribution of the thesis: a discrimninative approach to STD that extends previous work by Keshet et al. [2009]. Specifically, we develop an algorithm for discriminative STD that relaxes the constraint that sub-word state alignments be available for training query terms. The models are trained using a large-margin algorithm to optimize the expected area under the receiver operating characteristic (ROC) [Cortes and Mohri, 2004]. In order to determine the effectiveness of the proposed

[29] A version of the work described in this chapter has appeared previously in [Prabhavalkar et al., 2012, 2013].

[30] We use the terms *spoken term detection* and *keyword spotting* interchangeably in this thesis. Similarly we shall use the term *keyword* and *term* (in the context of term to be detected in speech) interchangeably.

approach in limited data settings, we conduct a systematic empirical evaluation in a simulated low-resource setting where training data are obtained by sampling limited subsets of utterances from the Switchboard dataset [Godfrey et al., 1992]. In experimental results, we find that the proposed discriminative STD systems outperform baseline hidden Markov model-based (HMM-based) acoustic STD systems [Szöke et al., 2005] across a range of training set sizes.

We begin in Section 4.1 by briefly describing previous work in STD and outline our motivation. In Section 4.2.1, we provide some intuition behind the proposed approach, that is then formalized in Section 4.3 where we formally introduce the proposed model for STD. We describe the algorithm for training the model in order to optimize expected area under the ROC curve in Section 4.4. We evaluate the proposed models against hidden Markov model-based (HMM-based) systems in Section 4.5. We conclude with a summary of the results in Section 4.6.

4.1 Background

The discussion of STD technology in the remainder of this section will be extremely brief; the goal of the subsequent discussion is to acquaint the reader with current STD technology and to motivate alternative paradigms.

The problem of detecting specific keywords in speech utterances is a well researched problem in the field of ASR. The earliest work in this area [Christiansen and Rushforth, 1977; Higgins and Wohlford, 1985] was based on extracting whole-word keyword templates from the training data and detecting the presence of the keywords using a dynamic time warping-based search. Subsequently, as hidden Markov models (HMMs) began to

dominate speech recognition technology, these models were also applied to the task of keyword spotting with good results: first in the form of whole-word HMMs [Wilpon et al., 1990] followed by sub-word HMMs [Rohlicek et al., 1989; Rose and Paul, 1990; Manos and Zue, 1997; Silaghi and Bourlard, 1999]. HMMs, in one form or another, have continued to remain the dominant paradigm in keyword spotting technology for the past couple of decades.

The current dominant paradigm for STD involves the use of trained large vocabulary continuous speech recognition (LVCSR) systems [Miller et al., 2007; Vergyri et al., 2007; Akbacak et al., 2008]. In such approaches, which we describe in detail in Chapter 6, the LVCSR system is used to decode speech utterances, to obtain a representation of the likely word sequences corresponding to the input speech (e.g., a word lattice or a confusion network [Mangu et al., 2000]). Each hypothesized word in the speech utterances can thus be associated with a score (e.g., the posterior probability of the word v given its hypothesized start and end time (s, e) in the utterance $\overline{\mathbf{x}}$: $P(v|s, e, \overline{\mathbf{x}})$) which can then be used to decide whether or not a particular word hypothesis should be declared as an occurrence of the search term by the system.

Although LVCSR-based approaches have been successful when applied at the task of STD, a significant limitation of such systems is that they typically involve a very large number of free parameters (e.g. in a recent Mandarin LVCSR system, Plahl et al. [2009] report that the system has on the order of $\sim 640M$ free parameters); robustly estimating these parameters requires the availability of a large amount of labeled training data, which may not always be available. For example, it would be desirable to be able to rapidly develop STD systems for low-resource languages or for porting existing STD systems to novel acoustic conditions.

The discriminative STD approach developed in this chapter, is one of a number of recent approaches for STD that have attempted to address the challenges of low-resource [Jansen and Niyogi, 2009; Karanasou et al., 2012] and zero-resource [Hazen et al., 2009; Zhang and Glass, 2009; Muscariello et al., 2011; Jansen and Durme, 2012; Norouzian et al., 2013] settings.

4.2 Notation and Preliminaries

The notation used in this chapter is consistent with the notation that we used in the previous chapter and it is briefly reviewed here; the reader may wish to consult Section 3.3 before proceeding.

We shall denote the parameterized acoustic speech signal (e.g., parameterized as a set of PLP coefficients) as $\overline{\mathbf{x}} \in \mathcal{X}^*; \overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$, $1 \leq t \leq T$, is a d-dimensional vector extracted from the t-th frame. We use the notation $\overline{v} \in \mathcal{V}^*$ to denote candidate terms: sequences of one or more words from the lexicon $\mathcal{V}$. We assume that we have access to a pronunciation dictionary, denoted by the function $\pi : \mathcal{V} \rightarrow \mathcal{P}^*$, where $\mathcal{P}$ is the set of phone symbols and $\mathcal{P}^*$ represents the set of all finite-length phone sequences. Thus, the pronunciation dictionary allows us to represent each word in terms of its corresponding phonetic representation.[31] We denote the number of phones in the canonical pronunciation of the term $\overline{v}$ as $|\overline{v}|$. Thus, $\pi(\overline{v}) = (\sigma_1, \sigma_2, \cdots, \sigma_{|\overline{v}|}) \in \mathcal{P}^*$. For example, for $\overline{v}$ = "sense", its canonical pronunciation is $\pi(\overline{v})$ = (s, eh, n, s), with $|\overline{v}| = 4$.

[31] By modeling the pronunciation dictionary as the function $\pi(\overline{v})$, we have implicitly assumed that every word sequence has a unique pronunciation, which is often not the case with most ASR systems. The CMU pronunciation dictionary [Weide, 2007], for example, lists two pronunciation of 'either' representing dialectical variation: /iy dh er/ and /ay dh er/. In our system, we assume that each word is assigned a single pronunciation, corresponding to the more likely pronunciation. In principle, it is straightforward to allow multiple pronunciations for the keywords although this is not done in our experiments and we leave this for future work.

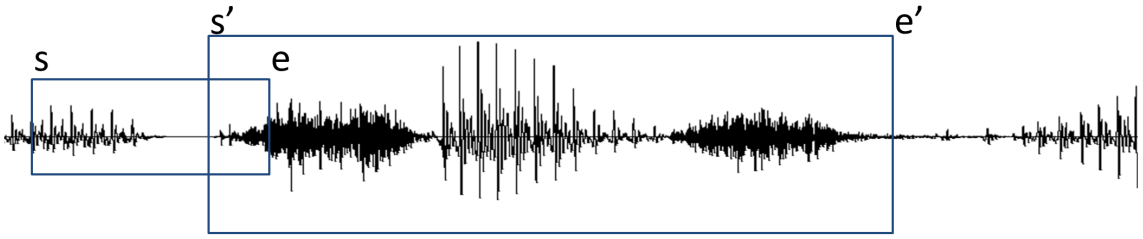

Figure 4.1: Intuition behind proposed model for Spoken Term Detection. (s, e) and (s', e') represent two of the $O(T^2)$ candidate search locations in the uttearance $\overline{\mathbf{x}}$. The model will be trained to produce higher scores for regions that are likely to correspond to the search term, and lower scores for regions that are unlikely to correspond to the search term.

As a final note, we make the additional assumption that a given search term $\overline{v}$ occurs at most once in a given speech utterance $\overline{\mathbf{x}}$; this allows us us to focus on the single highest-scoring region within the speech utterance as we describe in subsequent sections.[32]

4.2.1 Discriminative Spoken Term Detection: Intution

Before we formally describe the discriminative STD model, we begin by providing some intuition behind the proposed approach. As we have mentioned before, the goal of an STD system is to detect the location of a search term $\overline{v}$ within a speech utterance $\overline{\mathbf{x}}$, if present. To this end, we begin by considering every possible 'chunk' within the speech utterance, corresponding to contiguous sequences of speech frames, and evaluate these chunks with respect to whether or not they are likely to contain the term of interest. This is illustrated in Figure 4.1. In order to determine whether or not a given hypothesized start and end-time (s, e) , where $1 \leq s < e \leq T$, contains a given search term $\overline{v}$, we consider every possible segmentation of the phones corresponding to its phonetic pronunciation $\pi(\overline{v})$;

[32]This assumption is not particularly restrictive. If search terms can occur multiple times in the utterance $\overline{\mathbf{x}}$, we can extract multiple overlapping windows of speech frames within the utterance (the extent of the overlap would be based on the average durations of the phones in $\pi(\overline{v})$) and conduct a search in each such window of speech frames.

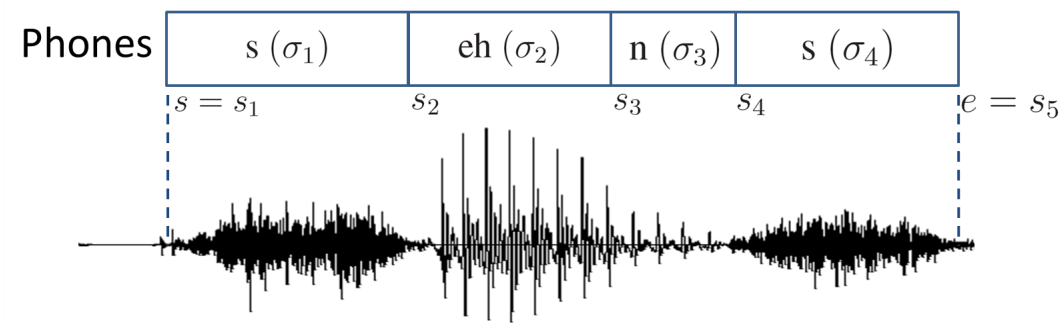

Figure 4.2: Schematic of the notation used in our discriminative STD model. For this example, $\overline{v}$ = "sense", $\pi(\overline{v})$ = (s, eh, n, s), with $|\overline{v}| = 4$. The figure illustrates one possible phonetic segmentation $\overline{\mathbf{s}}$ for a given start and end time (s,e).

intuitively, the segmentation of phones corresponding to the true segmentation should score highly while incorrect segmentations or segmentations in an incorrect hypothesized location (s, e) should not score highly.

Since the pronunciation is composed of $|\overline{v}|$ units, we represent a valid phone segmentation, as the vector $\overline{\mathbf{s}}$, which represents the sequence of start and end times for each of the phones in its pronunciation: $\overline{\mathbf{s}} = (s_1, s_2, \cdots, s_{|\overline{v}|})$, where the j-th unit in the pronunciation σ_j extends from frames s_j to $s_{j+1} - 1$, inclusive with $s_1 = s$ and $s_{|\overline{v}|+1} = e + 1$. We use the notation $\overline{\mathbf{s}}(\overline{v}) \sim (s, e)$ to denote a phonetic segmentation $\overline{\mathbf{s}}(\overline{v})$ that begins at frame s and ends at frame e. In order to simplify notation, we shall denote the phonetic segmentation as $\overline{\mathbf{s}}$, when the intended search term $\overline{v}$ is clear from context. Finally, we shall denote the phoneme hypothesized at time t under segmentation $\overline{\mathbf{s}}$ as $p_t(\overline{\mathbf{s}})$, i.e., $p_t(\overline{\mathbf{s}}) = \sigma_j$ for $s_j \leq t < s_{j+1}$. Our notation appears in Figure 4.2.

4.3 Model for Discriminative Spoken Term Detection

In this section, we formalize our intuitions from Section 4.2.1 by describing how we construct a function for discriminative spoken term detection extending previous work by Keshet et al. [2009]. Our goal is to learn a function $f : \mathcal{X}^* \times \mathcal{V}^* \rightarrow \mathbb{R}$, which takes as its input a speech utterance $\overline{\mathbf{x}} \in \mathcal{X}^*$ and a query term $\overline{v} \in \mathcal{V}^*$, and returns a score $f(\overline{\mathbf{x}}, \overline{v}) \in \mathbb{R}$ representing the confidence that the query term occurs in the utterance. In a practical system, the utterance $\overline{\mathbf{x}}$ would be declared a putative hit for a query term $\overline{v}$ if $f(\overline{\mathbf{x}}, \overline{v}) > b(\overline{v})$ for some (user-modifiable) threshold $b(\overline{v}) \in \mathbb{R}$. We model the STD function, parameterized by a set of linear weights $\mathbf{w} \in \mathbb{R}^n$, as

$$f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{v}) = \max_{\overline{\mathbf{s}} \in \mathcal{S}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}) \tag{4.1}$$

where $\mathcal{S}$ is the set of all valid articulatory segmentations over all possible start and end times in the utterance: $\mathcal{S} = \{\overline{\mathbf{s}}(\overline{v}) : \overline{\mathbf{s}}(\overline{v}) \sim (s, e) \text{ where } 1 \leq s \leq e \leq T\}$, and $\boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}) \in \mathbb{R}^n$ is a feature vector. The score in Equation 4.1 corresponds to the score of the highest scoring segmentation, $\overline{\mathbf{s}}$, over all possible start and end times within the utterance $\overline{\mathbf{x}}$ for the term $\overline{v}$.

In our work, the feature vectors, $\boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}})$, are composed of a set of pre-defined feature maps $\{\phi_j\}_{j=1}^m$, where $\phi_j : \mathcal{X}^* \times \mathcal{V}^* \times \mathcal{S} \rightarrow \mathbb{R}^r$. Each feature map takes as input the acoustics $\overline{\mathbf{x}}$, the term $\overline{v}$, and the articulatory segmentation $\overline{\mathbf{s}}$ and returns an r-dimensional vector. The specific form of the feature maps used in our experiments is described in Section 4.3.1.

Naively computing the maximization in Equation 4.1 by explicitly considering every possible segmentation would necessitate a search over $O(T^{|\overline{v}|})$ segmentations, which would be prohibitively slow for terms with large $|\overline{v}|$. However, in the case where the feature maps can be decomposed into a form that exhibits optimal substructure, the maximizing

segmentation can be computed using dynamic programming as described in [Prabhavalkar et al., 2011].[33]

4.3.1 Feature Maps

In this section, we describe two types of feature maps used that we use in our system. Our feature maps are constructed using a set of feature functions, $\boldsymbol{\xi} : \mathcal{X} \rightarrow \mathbb{R}^r$, that are constructed from the acoustic feature vectors which allows the system to incorporate information from diverse sources. We denote the feature functions as a vector-valued function $\boldsymbol{\xi} : \mathcal{X} \rightarrow \mathbb{R}^r$, which takes as input an acoustic feature vector corresponding to a frame of speech $\mathbf{x} \in \mathcal{X}$ and outputs a vector in $\mathbb{R}^r$. In our experiments, these are constructed from multilayer perceptron detectors of phones and articulatory features.

The first set of feature maps computes the confidence that the acoustic frames correspond to the phoneme hypothesized at each frame in a given segmentation corresponding to the target term:

$$\boldsymbol{\phi}_{1,q} = \frac{1}{e-s+1} \sum_{t=s}^{e} \boldsymbol{\xi}(\mathbf{x}_t) \delta[p_t(\bar{\mathbf{s}}) = q] \tag{4.2}$$

where $q \in \mathcal{P}$ represents a particular phone. Thus, we have a set of $|\mathcal{P}|$ feature maps, each of which is a vector-valued function of the same length as $\boldsymbol{\xi}$.

The second set of feature maps model the acoustics at phone transitions for each pair of phones $q, q' \in \mathcal{P}$:

$$\boldsymbol{\phi}_{2,q,q'} = \frac{1}{e-s+1} \sum_{t=s+1}^{e} \boldsymbol{\xi}(\mathbf{x}_t) \delta[p_{t-1}(\bar{\mathbf{s}}) = q \wedge p_t(\bar{\mathbf{s}}) = q'] \tag{4.3}$$

Thus, we have a total of $|\mathcal{P}|^2$ feature maps of the second type, for each pair of phones, each of which is a vector-valued function of the same length as $\boldsymbol{\xi}$. We note that in both

[33] This corresponds to running the 'max-product' version of the 'sum-product' algorithm presented in Figure 3.9, wherein the summation operation in the alpha-beta recurrences is replaced by the 'max' operation. This would compute the highest scoring SubwordConfiguration for the term $\overline{v}$ corresponding to a given start and end time.

Equations 4.2 and 4.3, we normalize the feature maps by the length in frames, $(e-s+1)$, of the hypothesized segmentations $\overline{\mathbf{s}}$. This is done in order to ensure that the scores computed across segmentations of different length are comparable to each other.

4.4 Training the Model to Optimize Area Under the Receiver Operating Characteristic

Our goal is for the detector to be able to detect any term $\overline{v}$ in the test set, including those terms that may not have been seen in training, as long as pronunciation, $\pi(\overline{v})$, for the term is available. In what follows, we also assume that the input $\overline{\mathbf{x}}$ is an utterance short enough for any term of interest to occur at most *once*. As we mentioned before, this is not a restrictive assumption, since for longer signals, the detector may be applied in a sliding window of appropriate length on overlapping portions of the utterance. We now describe a discriminative algorithm for learning the parameters $\mathbf{w}$ of the model presented in Equation 4.1 using a set of paired training examples that optimizes the area under the receiver operating characteristic (ROC).

4.4.1 Area Under the Receiver Operating Characteristic

The performance of a STD system, is often measured in terms of the receiver operating characteristic (ROC). The ROC is obtained by sweeping the decision threshold from $(-\infty, \infty)$ and plotting the true-positive rate (detection rate; fraction of positive examples that are correctly classified, i.e. score above the threshold) versus the false-positive rate (fraction of negative examples that are incorrectly classified, i.e. score above the threshold) over the entire range. This is illustrated in Figure 4.3. Each point on the curve represents a particular operating point of the system. A single metric that describes system performance, the average performance over all operating points, can be computed as the area under the

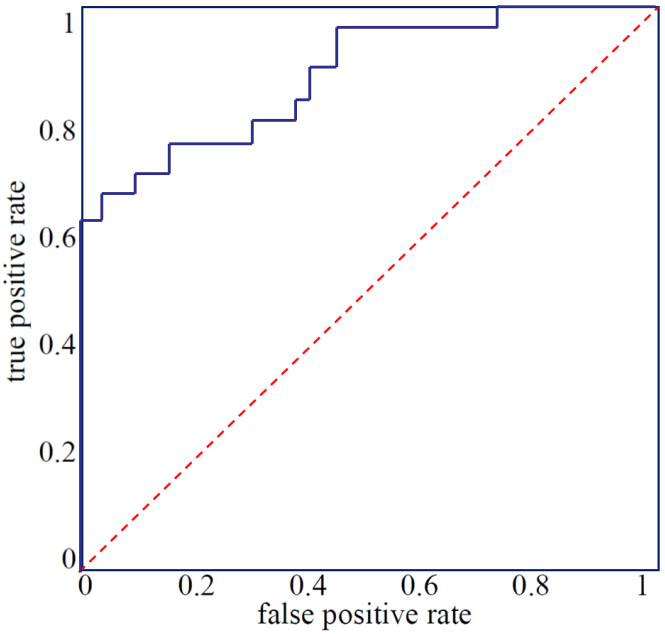

Figure 4.3: An example of an ROC curve for a system.

ROC curve (AUC). Independent of the proportion of positive and negative examples, the AUC of a system ranges from 0.5 (chance performance) to 1.0 (perfect detection).

In the next section, we propose an algorithm for optimizing expected AUC on unseen terms by extending the algorithm in [Keshet et al., 2009]. The method presented here can be adapted to other evaluation functions, such as the *occurrence-weighted value* or the *actual term-weighted value (ATWV)* used in the 2006 NIST STD evaluation [Fiscus et al., 2007] or the Figure of Merit (FoM) [Wallace et al., 2011].

4.4.2 Training to Optimize Expected AUC

Our goal is to find the weight vector, $\mathbf{w}^*$, that maximizes the expected AUC for unseen query terms. More formally, assume that we draw a triplet, $(\overline{v}, \overline{\mathbf{x}}^+, \overline{\mathbf{x}}^-)$, from a fixed but unknown distribution ρ, where $\overline{\mathbf{x}}^+$ and $\overline{\mathbf{x}}^-$ represent utterances in which the term $\overline{v}$ is either present or absent respectively. The optimal weight vector can be represented in terms of

the *Wilcoxon-Mann-Whitney statistic* [Cortes and Mohri, 2004] as,

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmax}}\ \mathbb{P}\Big[f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) > f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})\Big] \tag{4.4}$$

$$= \underset{\mathbf{w}}{\text{argmax}}\ \mathbb{E}\Big[\delta[f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) > f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})]\Big] \tag{4.5}$$

where the probability and expectation are computed with respect to $(\overline{v}, \overline{\mathbf{x}}^+, \overline{\mathbf{x}}^-) \sim \rho$. Since this distribution is unknown, we approximate it using the empirical distribution corresponding to the training set, $\mathcal{T}$, of N examples drawn from the same probability distribution,

$$\mathcal{T} = \{\overline{v}_i, \overline{\mathbf{x}}_i^+, \overline{\mathbf{x}}_i^-, s_i^+, e_i^+\}_{i=1}^N \tag{4.6}$$

where, each training example consists of the query term $\overline{v}_i \in \mathcal{V}^*$, an utterance $\overline{\mathbf{x}}_i^+ \in \mathcal{X}_{\overline{v}_i}^+$ in which the term $\overline{v}_i$ is uttered (*a positive utterance*), an utterance $\overline{\mathbf{x}}_i^- \in \mathcal{X}_{\overline{v}_i}^-$ in which the term $\overline{v}_i$ is not uttered (*a negative utterance*), and the start and end frames (s_i^+, e_i^+) corresponding to the location of the query term in the positive utterance.

Since the sets of positive and negative utterances are not required to be disjoint for different query terms, the same utterance may represent a positive example for some term, $\overline{v}_i$, while simultaneously serving as a negative example for another term, $\overline{v}_j$ $(i \neq j)$. The ability to create multiple training examples, corresponding to different query terms from the same utterance, allows for efficient use of limited training data.

Maximizing the AUC is equivalent to minimizing the expectation over $\delta[f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) < f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})]$. This, in turn, is equivalent to minimizing the expectation over $\delta[f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v}) - f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) > 0]$. The structural hinge-loss is an upper bound to this term, and is defined as

$$\ell(\overline{v}, \overline{\mathbf{x}}^+, \overline{\mathbf{x}}^-, \mathbf{w}) = \big[1 - f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) + f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})\big]_+ \tag{4.7}$$

where $[z]_+ = \max\{0, z\}$. Thus, the weight vector $\mathbf{w}^*$ can be found by minimizing the following regularized average structural hinge-loss over the training set:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbf{f}^{\text{org}}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left[1 - f_{\mathbf{w}}(\overline{\mathbf{x}}_i^+, \overline{v}_i) + f_{\mathbf{w}}(\overline{\mathbf{x}}_i^-, \overline{v}_i)\right]_+ \quad (4.8)$$

Note that in Equation 4.8, in computing $f_{\mathbf{w}}(\overline{\mathbf{x}}_i^+, \overline{v}_i)$, we restrict the search to only those segmentations $\overline{\mathbf{s}}$, that begin and end at the appropriate times corresponding to the location of the search term in the positive utterance: $f_{\mathbf{w}}(\overline{\mathbf{x}}_i^+, \overline{v}_i) = \max_{\overline{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}})$. In computing $f_{\mathbf{w}}(\overline{\mathbf{x}}_i^-, \overline{v}_i)$, however, we search over all possible start and end times within the utterance. We note that the the formulation of the optimization problem in Equation 4.8 differs from the approach in [Keshet et al., 2009] in that we do not assume that the *true* segmentation ($\overline{\mathbf{s}}_i^+$) of the terms in the positive utterances is known; instead we seek to implicitly determine this information as part of the training procedure. This is particularly useful for the experiments that appear in the next chapter, where we replace the phone-based pronunciation model with an articulatory feature-based pronunciation model; obtaining the *true* articulatory segmentations is significantly harder than obtaining phonetic segmentations. The impact of this choice is investigated in experiments presented in Section 4.5.2.

4.4.3 Solving the Non-Convex Optimization Problem in Equation 4.8 using the Majorization-Minimization Algorithm

The optimization problem that appears in Equation 4.8, is a non-convex optimization problem (because of the presence of the term $-\max_{\overline{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}})$ that appears in Equation 4.8). Instead of solving the optimization problem directly, we use the Majorization-Minimization (MM) algorithm [Hunter and Lange, 2004].[34]

[34]The algorithm that we use to optimize Equation 4.8 was described as an instance of the convex-concave procedure (CCCP) [Yuille and Rangarajan, 2002] in our previous work [Prabhavalkar et al., 2012, 2013]. Although it is possible to view our algorithm (which appears in Figure 4.5) as an instance of CCCP (with

The Majorization-Minimization Algorithm

The MM algorithm [Hunter and Lange, 2004] is a conceptually simple iterative procedure for minimizing a function $f(\boldsymbol{\theta})$, given an estimate of the minimizer $\boldsymbol{\theta}_m$ at the mth iteration. In fact, a number of algorithms, such as the E-M algorithm [Dempster et al., 1977] and the convex-concave procedure (CCCP) [Yuille and Rangarajan, 2002] can be shown to be special cases of this algorithm. The main idea behind the algorithm is that instead of minimizing the function $f(\boldsymbol{\theta})$ directly, the algorithm proceeds by first constructing a function $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$, known as the majorizer (at $\boldsymbol{\theta}_m$). A function $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ is said to *majorize* the function $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_m$ if it satisfies the following two conditions:

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}_m) \geq f(\boldsymbol{\theta}) \quad \text{for all } \boldsymbol{\theta} \tag{4.9}$$

$$g(\boldsymbol{\theta}_m; \boldsymbol{\theta}_m) = f(\boldsymbol{\theta}_m) \tag{4.10}$$

Intuitively, the surface of the majorizer $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$, lies above the surface of the original function $f(\boldsymbol{\theta})$ at all points in the parameter space. This observation equips us with a simple iterative procedure to solve the original minimization problem: We simply minimize the majorizer $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ instead to obtain a new estimate $\boldsymbol{\theta}_{m+1}$. This minimizer of $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ represents an improved estimate of the minimizer of $f(\boldsymbol{\theta})$ since,

$$g(\boldsymbol{\theta}_{m+1}; \boldsymbol{\theta}_m) \leq g(\boldsymbol{\theta}_m; \boldsymbol{\theta}_m) \qquad (\text{since } \boldsymbol{\theta}_{m+1} \text{ minimizes } g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)) \tag{4.11}$$

$$f(\boldsymbol{\theta}_{m+1}) \leq g(\boldsymbol{\theta}_{m+1}; \boldsymbol{\theta}_m) \qquad (\text{from Eq. } 4.11) \tag{4.12}$$

$$f(\boldsymbol{\theta}_{m+1}) \leq f(\boldsymbol{\theta}_m) \qquad (\text{from Eq. } 4.10, 4.11, 4.12) \tag{4.13}$$

The steps involved in a single iteration of the MM algorithm are illustrated graphically in Figure 4.4.

a certain approximation), it is more readily and intuitively seen to be an instance of the MM algorithm (of which CCCP is a special case). We shall therefore refer to the algorithm as an instance of the MM algorithm in this section. We refer the interested reader to Appendix A where the connection to CCCP is made explicit.

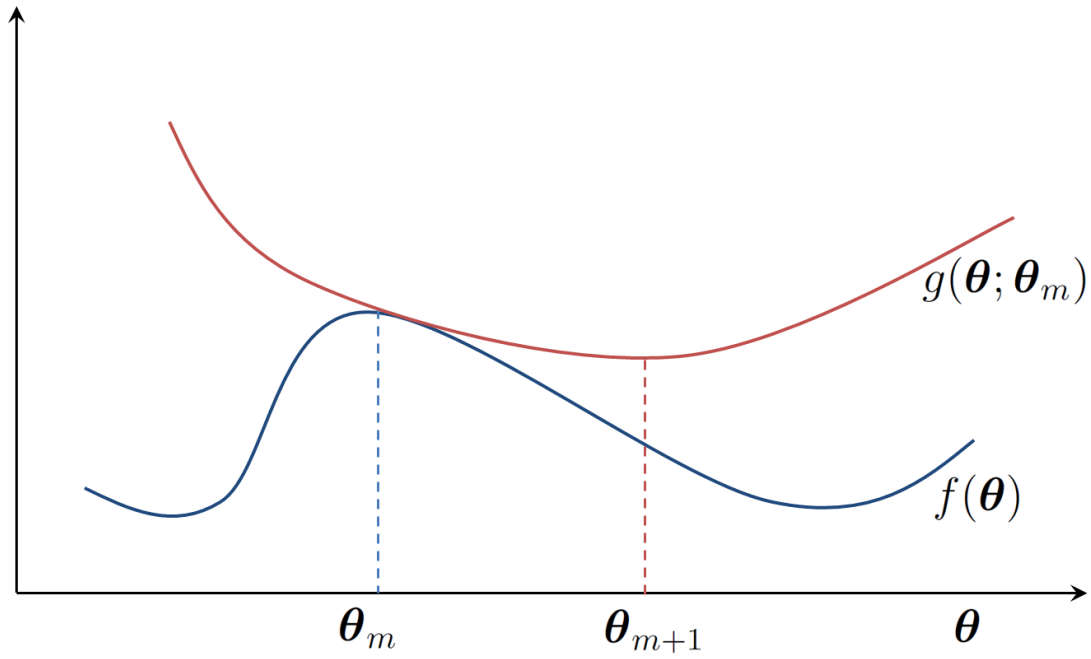

Figure 4.4: An illustration of the MM algorithm [Hunter and Lange, 2004]. Given an initial estimate, $\boldsymbol{\theta}_m$, of the minimizer of $f(\boldsymbol{\theta})$, the algorithm begins by constructing the majorizer $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ of $f(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta}_m$. The surface of the majorizer, $g(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$, touches $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_m$ and lies above the original function at all other points. If $\boldsymbol{\theta}_{m+1}$ represents a point that corresponds to a lower value of the majorizer than $\boldsymbol{\theta}_m$, then $f(\boldsymbol{\theta}_{m+1}) \leq f(\boldsymbol{\theta}_m)$. Thus, the MM algorithm iteratively converges to a local minimum of the original function $f(\boldsymbol{\theta})$.

4.4.4 Using the MM Algorithm for Minimizing Equation 4.8

In order to optimize Equation 4.8, given an estimate $\mathbf{w}_t$, we describe the creation of a majorization function, which can be optimized using the MM algorithm. The first observation to be made is that the structural hinge loss, $\ell(\overline{v}, \overline{\mathbf{x}}^+, \overline{\mathbf{x}}^-, \mathbf{w})$, that appears in the optimization function can be upper-bounded as follows:

$$\ell(\overline{v}, \overline{\mathbf{x}}^+, \overline{\mathbf{x}}^-, \mathbf{w}) = \left[1 - f_{\mathbf{w}}(\overline{\mathbf{x}}^+, \overline{v}) + f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})\right]_+ \tag{4.14}$$

$$= \left[1 - \max_{\overline{\mathbf{s}} \sim (s^+, e^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}^+, \overline{v}, \overline{\mathbf{s}}) + f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})\right]_+ \tag{4.15}$$

$$\leq \left[1 - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}^+, \overline{v}, \overline{\mathbf{s}}_0) + f_{\mathbf{w}}(\overline{\mathbf{x}}^-, \overline{v})\right]_+ \tag{4.16}$$

where the inequality in Equation 4.16 holds for any fixed $\overline{\mathbf{s}}_0 \sim (s^+, e^+)$.

In order to define the majorizer at $\mathbf{w}_t$, for $\mathbf{f}^{\text{org}}(\mathbf{w})$, we compute for each training example, $\overline{\mathbf{s}}_i^+(\mathbf{w}_t)$ as,

$$\overline{\mathbf{s}}_i^+(\mathbf{w}_t) = \operatorname*{argmax}_{\overline{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w}_t \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}) \tag{4.17}$$

Thus, $\overline{\mathbf{s}}_i^+(\mathbf{w}_t)$ represents the segmentation that results in the highest score at the current estimate $\mathbf{w}_t$ of the weights. Now define, $g^{\text{maj}}(\mathbf{w}; \mathbf{w}_t)$ by replacing the original segmentations for positive examples that appear in the optimization problem with the computed segmentation from Equation 4.17,

$$g^{\text{maj}}(\mathbf{w}; \mathbf{w}_t) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left[1 - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+(\mathbf{w}_t)) + f_{\mathbf{w}}(\overline{\mathbf{x}}_i^-, \overline{v}_i)\right]_+ \tag{4.18}$$

The function $g^{\text{maj}}(\mathbf{w}; \mathbf{w}_t)$ in Equation 4.18 is a majorizer of $\mathbf{f}^{\text{org}}(\mathbf{w})$ at $\mathbf{w}_t$ and can be minimized using the MM algorithm. This follows since $g^{\text{maj}}(\mathbf{w}; \mathbf{w}_t) \geq \mathbf{f}^{\text{org}}(\mathbf{w})$ by Equation 4.16 and $g^{\text{maj}}(\mathbf{w}_t; \mathbf{w}_t) = \mathbf{f}^{\text{org}}(\mathbf{w}_t)$ by definition.

Finally, notice that the optimization problem of minimizing $g^{\text{maj}}(\mathbf{w}; \mathbf{w}_t)$ – the 'inner loop' in the MM algorithm – is exactly the same as the optimization problem solved by

Input: training set $\mathcal{T} = \{\overline{v}_i, \overline{\mathbf{x}}_i^+, \overline{\mathbf{x}}_i^-, s_i^+, e_i^+\}_{i=1}^m$; parameter λ

Initialize: $\mathbf{w}_0 = \mathbf{0}$

For $t = 0, \ldots, T-1$

  For $i = 1, \ldots, m$

    Predict: $\overline{\mathbf{s}}_i^+ = \text{argmax}_{\overline{\mathbf{s}} \sim (s_i^+, e_i^+)} \ \mathbf{w}_t \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}})$

  Set: $\mathbf{u}_0 = \mathbf{w}_t$

  For $j = 0, \ldots, J-1$

    Pick example $(\overline{v}_i, \overline{\mathbf{x}}_i^+, \overline{\mathbf{x}}_i^-, \overline{\mathbf{s}}_i^+)$, $1 \leq i \leq m$

    Predict: $\overline{\mathbf{s}}_i^- = \text{argmax}_{\overline{\mathbf{s}}} \ \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})$

    Set: $\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-)$

    Set: $\alpha_i = \min \left\{ \frac{1}{\lambda}, \frac{[1 - \mathbf{u}_j \cdot \Delta\boldsymbol{\phi}_i]_+}{\|\Delta\boldsymbol{\phi}_i\|^2} \right\}$

    Update: $\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_i \Delta\boldsymbol{\phi}_i$

  Update: $\mathbf{w}_{t+1} = \frac{1}{J} \sum_{j=1}^J \mathbf{u}_j$

Output: The last weight $\mathbf{w}_T$.

Figure 4.5: Majorization-Minimization (MM) algorithm to optimize Equation 4.8. The algorithm uses the passive-aggressive algorithm [Crammer et al., 2006] as the inner loop to minimize the majorizer [Keshet et al., 2009]. Details of the derivation of the passive-aggressive update for our problem can be found in Appendix B.

Keshet et al. [2009] and can be solved in an online fashion using the passive-aggressive algorithm [Crammer et al., 2006]. Pseudocode of the training algorithm is given in Figure 4.5. Details of the derivation of the passive-aggressive update for our problem appear in Appendix B.

4.5 Experiments on the Switchboard Corpus

In order to determine the effectiveness of the proposed models, we conduct experiments on a subset of the Godfrey et al. [1992], using varying training set sizes, to determine how performance of the proposed algorithm varies as a function of the amount of training data. We begin by selecting as our candidate set of utterances all sentences in Switchboard sets 23–49 containing at least four words other than non-speech sounds;[35] candidate sentences can contain non-speech sounds as long as they contain at least four words in addtion to the non-word tokens. From this candidate set, we build four training corpora of increasing size containing 500, 1000, 2500, and 5000 sentences, such that each corpus is included in the next larger set. In other words, all data contained in a set of smaller size is also contained in the sets of larger size. We construct a 40-keyword set for parameter tuning and a 60-keyword set for final testing by selecting words from sets 20–22 that occur at least five times in Switchboard and contain at least five phonemes in their canonical pronunciations. For each keyword, we select 20 sentences containing the keyword (positive sentences) and 20 sentences not containing the keyword (negative sentences) to obtain corresponding development and test sets. We remove initial and final silences from all utterances in the train, development and test sets.[36]

Generation of Training Examples

For each sentence in a training corpus, we select each word v_i that contains at least 5 phonemes in its canonical pronunciation as a candidate term, and we select the corresponding utterance as an instance of a positive example $\overline{\mathbf{x}}_i^+$ for that term. We randomly select a

[35] By non-speech sounds we refer to noise, silence, fragments, and laughter.

[36] Details of the utterances corresponding to the train, development and test sets and the corresponding development and test keywords are available at http://ttic.uchicago.edu/~jkeshet/Keyword_Spotting.html

Metric	500	1000	2500	5000
Training Data (hrs)	0.8	1.5	3.7	7.4
Generated Positive Examples	1538	2876	7245	14570

Table 4.1: Statistics for the four training datasets chosen by sub-selecting utterances from Switchboard [Godfrey et al., 1992] used in our experiments.

sentence from the training corpus that does not contain the keyword as a negative example $\overline{\mathbf{x}_i}$. The set so selected serves as a training set for the discriminative spoken term detection systems. In order to be comparable to the baseline systems described shortly, we model each phone in pronunciation of the term using 3-state models. The statistics of the data sets used in our experiments appear in Table 4.1.

Computation of Feature Functions: Tandem Feature Generation

We compute the functions $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_r]$ following the basic methodology outlined in [Prabhavalkar et al., 2011]. We train four multilayer perceptrons (MLPs), three of which are frame classifiers of articulatory features: lip configuration (L, 8 labels), tongue configuration (T, 25 labels), and glottis-velum (G, 5 labels). The final MLP is a frame-level classifier of phones. Unlike the work in [Prabhavalkar et al., 2011], these MLPs are trained on all phonetically transcribed data from sets 23–49 of the Switchboard Transcription Project (STP) data [Greenberg et al., 1996] using the Quicknet toolkit [Johnson et al., 2004].

We parameterize the acoustics using 12^{th}-order PLP coefficients with energy, deltas and double-deltas to obtain a 39-dimensional input representation. The feature vectors for a given frame are concatenated with the four preceding and succeeding frames to obtain a 351-dimensional input representation to the MLPs. The MLPs are single hidden layer feed-forward nets, with a sigmoid activation function on hidden layer nodes and a softmax

output function on the output layer nodes, and are trained to optimize a cross-entropy criterion. The number of hidden nodes is tuned on a held-out development set. Once the MLPs are trained, we compute log-posteriors for the data in the training, development and test sets for all four MLPs and project all of these log-posteriors down to their top 39 principal components using Principal Components Analysis (PCA) to obtain a *tandem feature* representation Hermansky et al. [2000]. These features serve as the observations modeled using a mixture of Gaussians in our baseline GMM-HMM systems and are also used in the feature functions $\boldsymbol{\xi}$ of the discriminative systems (after the incorporation of a constant 1 to the vector to model a bias term, so that $r = 40$). We model the pronunciations of words using 3 states per phone label.

Baseline GMM-HMM systems

The proposed systems are evaluated against HMM-based acoustic keyword spotting systems [Szöke et al., 2005] that are trained using HTK [Young et al., 2002]. The baselines are constructed by defining a recognition network consisting of a keword network – created by concatenating together 3-state HMM phone models corresponding to the pronunciation of the term $\pi(\overline{v})$ – in parallel with a garbage network consisting of all phone models in parallel. We consider two baseline systems which differ in how the keyword network is modeled: either using (a.) context-independent monophones (HMM-mono) or (b.) context-dependent word-internal triphones (HMM-tri). In both baselines, the garbage network is modeled using context-independent monophones for computational efficiency.

Given a test utterance, we compute the one-best Viterbi path through the network, which either passes through the keyword model (a detection) or passes solely through the garbage model (a non-detection). The trade-off between the true positive and false positive rates is set by varying the keyword insertion probability. By varying the term insertion probability,

System	500	1000	2500	5000
HMM-PLP-mono	0.773	0.811	0.849	0.857
HMM-mono	0.810	0.827	0.846	0.857
HMM-tri	0.828	0.855	0.899	0.920
Disc-Phone	0.874^*	0.901^*	0.917	0.933^*

Table 4.2: Test set average AUC for the baseline HMM-based system and the proposed discriminative system. (*) indicates a significant ($p \leq 0.05$) improvement over the triphone HMM baseline using a one-tailed wilcoxon signed ranks test. The discriminative phone-based system significantly ($p \leq 0.001$) outperforms both monophone HMM baselines for all training set sizes.

we can generate the ROC, and therefore the AUC, for each term. We also report results on a monophone HMM-baseline (HMM-PLP-mono) that models the acoustics directly in terms of PLP coefficients (12th order with energy, deltas and double-deltas) to evaluate the effect of using 'tandem features' instead of PLP in the HMM-mono and HMM-tri baselines. The recognition network used in the baselines is illustrated in Figure 4.6. The number of Gaussian components per mixture was tuned separately for each set based on performance on the development set. The monophone baseline HMM system trained on the 500 sentences employed 32 Gaussian components per mixture; the system trained on 1000 examples employed 64 Gaussian components per mixture while the systems trained on 2500 and 5000 utterances used 128 Gaussian components per mixture since in pilot experiments, adding additional Gaussian components did not result in significant performance improvements. The triphone-based HMM systems employ either 8 (1000, 2500 sets) or 16 Gaussians (500, 5000 sets).

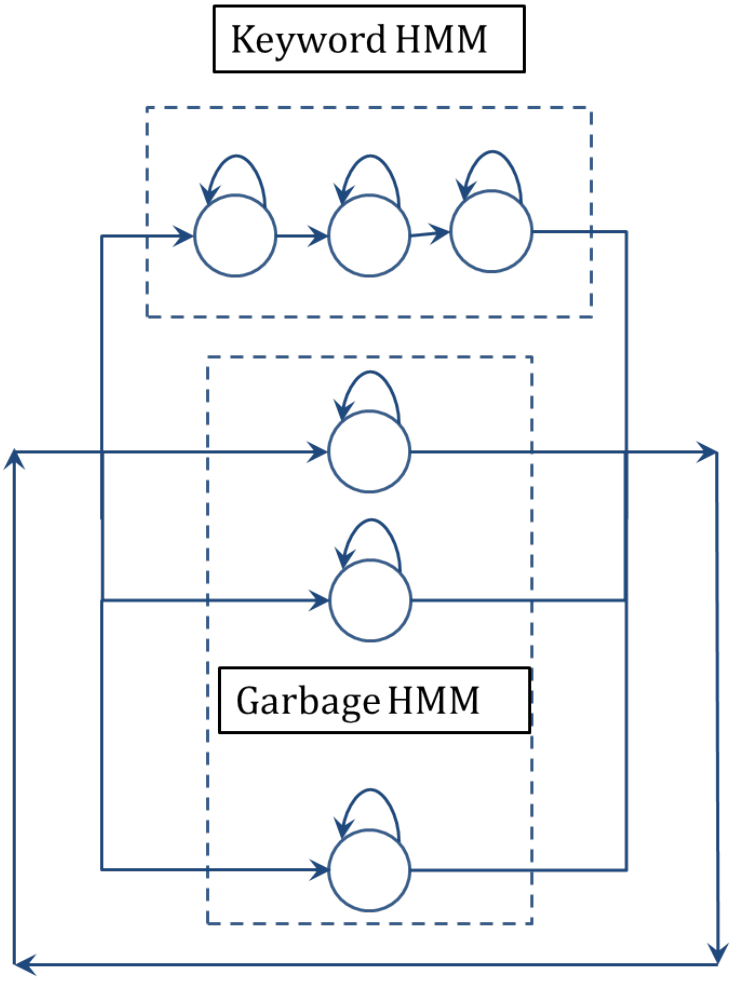

Figure 4.6: Baseline HMM spoken term detection system [Szöke et al., 2005].

4.5.1 Results I: Comparison of Performance of Proposed Discriminative System Against Baselines

Table 4.2 shows results in terms of AUC performance averaged across all terms in the test set. A number of interesting observations can be made from the results in the table. The first observation is that the use of the 'tandem features' in HMM-mono, improves performance over the PLP monophone baseline HMM-PLP-mono, in the lowest data cases of 500 and 1000 utterances; as the amount of data increases, the two systems perform identically. This observation might be explained by the fact that the discriminative 'tandem' features are trained on approximately one hour of additional data; however, as the amount of training data increases, these relative gain obtained from this additional data is diminished.

The second observation, is that the discrimnative system outperforms the monophone-based HMM-systems at all training set sizes by large margins; the performance of the discriminative system is significantly better than the monophone HMM baselines ($p \leq 0.001$) using a one-tailed Wilcoxon signed-rank test[37] across all training set sizes. Perhaps more surprisingly, although the performance of both the HMM-based and discriminative systems improves with increasing training set size, the discriminative systems even at very low training set sizes performs comparably to the monophone HMM baselines trained on much larger data sets. This is indicative of the fact that the discriminative systems, with significantly fewer parameters than the baseline HMMs are able to more effectively utilize the available training data.

[37]The Wilcoxon signed-rank test is a non-parametric test that compares the differences between paired samples (in our case, the individual AUCs for each term computed using the two methods under test). Thus, small differences between the paired samples can result in highly significant differences overall, even if the mean AUCs from the two systems do not differ by a large amount.

Finally, when we compare performance of the discriminative phone-based system against the stronger triphone-baseline, we find that the discriminative system significantly ($p \leq 0.05$) outperforms the tri-phone baseline at all training set sizes except for the set with 2500 utterances ($p = 0.062$). This is particularly encouraging, because our discriminative systems are context-independent. It is fairly straightforward to add context dependence to our discriminative models; we leave this as future work.

4.5.2 Results II: Comparison of Proposed Algorithm Against Model Proposed in [Keshet et al., 2009]

In our second set of experiments on the Switchboard dataset, we conduct experiments to quantify the effect of treating the segmentations $\overline{\mathbf{s}}_i^+$ of the positive terms in the training examples as unknown as opposed to the algorithm of Keshet et al. [2009] where the segmentations are assumed to be known and fixed. In order to determine the phoneme segmentations for the keywords $\overline{v}_i$ in the positive utterances $\overline{\mathbf{x}}_i^+$, we use the baseline monophone tandem HMM system to generate phoneme forced-alignments for each of the positive keyword examples that appear in the training set. Once the phoneme segmentations have been generated, we train a system using the discriminative algorithm described in Figure 4.5 except that we treat the segmentations $\overline{\mathbf{s}}_i^+$ as fixed, and do not re-compute them on each pass through the dataset thus implementing the algorithm in [Keshet et al., 2009], which we refer to as Disc-FixedSeg.

We also conduct an experiment, where we fully train the system assuming fixed and known segmentations (i.e., the fully trained Disc-FixedSeg system), and then use the weights learned in this system as an initialization of the weights in a new system. We then employ the algorithm described in Figure 4.5 to update the weights in the new system. In other words, we begin by assuming fixed segmentations; once the system is fully trained we

System	500	1000	2500	5000
Disc-Phone	0.874	0.901	0.917	0.933
Disc-FixedSeg	0.900*	0.904	0.928*	0.937
Disc-FixedSegInit	0.900	0.904	0.928	0.942*†

Table 4.3: Test set average AUC for proposed discriminative system compared against the algorithm of [Keshet et al., 2009]. Results marked (*) represent significant differences ($p \leq 0.05$) using a one-tailed Wilcoxon signed-ranks test over Disc-Phone. Results marked (†) represent a significant difference ($p \leq 0.05$) over Disc-FixedSeg using a one-tailed Wilcoxon signed-ranks test.

allow the segmentations to be modified if this allows for improvements in system performance. We refer to this system as Disc-FixedSegInit. Our results are presented in Table 4.3.

As can be seen in Table 4.3, using fixed segmentations for training the system (Disc-FixedSeg) produces significant improvements over the system that treats the segmentations as unknown (Disc-Phone) for two of the training set sizes (500, 2500) while there is no significant difference between the systems in the other two cases (1000, 5000) utterances. However, using the trained Disc-FixedSeg system to initialize our systems leads to improvements on the development set but not test sets for three of the training set sizes (500, 1000, 2500) and a significant improvement for the largest dataset size (5000).

Although the results are inconclusive, it appears that at least in the largest data setting, the algorithm proposed in this work is not significantly different in terms of performance from the algorithm proposed in [Keshet et al., 2009]. The data also indicate that peformance of the algorithm in this work might be sensitive to initialization (this is not extremely surprising, since the problem is a non-convex optimization problem). It is nevertheless encouraging that a system that does not have access to the true phone segmentations performs about as well as the system that utilizes these segmentations, since in the next chapter

(Chapter 5) we incorporate an articulatory feature-based pronunciation model, where it is more difficult to get access to the ground truth articulatory segmentations.

4.6 Summary

In this chapter, we presented a discriminative algorithm for STD that extends previous work [Keshet et al., 2009] and evaluated the algorithm in a setting of limited training data, simulated by selecting utterances from the Switchboard [Godfrey et al., 1992] dataset. In experimental results, we found that the proposed approach results in significant gains over baseline GMM-HMM systems across a range of training set sizes. In comparisons against the algorithm proposed in [Keshet et al., 2009], we found that the proposed algorithm performed slightly worse in some training set sizes, but that algorithm performance could be improved by suitable initialization.

CHAPTER 5: DISCRIMINATIVE SPOKEN TERM DETECTION WITH ARTICULATORY FEATURE-BASED PRONUNCIATION MODELS

The experiments presented in Chapter 4 demonstrated the effectiveness of the proposed discriminative spoken term detection (STD) systems in the setting of limited training data. In experiments on subsets of the Switchboard [Godfrey et al., 1992] dataset, the proposed approach outperformed the baselines across a range of training set sizes. In this chapter, we investigate discriminative models for STD that incorporate an articulatory feature-based (AF-based) pronunciation model. These models are aimed at better accounting for the pronunciation variation observed in conversational speech. The AF-based pronunciation models are similar to those described in Chapter 3 with one major difference: instead of treating the articulatory feature labels as known at training time, in this chapter we treat the articulatory feature streams as latent variables and allow the training data to guide the model in iteratively determining the optimal articulatory feature alignments. Removing the restriction that articulatory feature targets be known beforehand allows for rapid development of STD systems and is particularly well suited to the low-resource settings that we motivated in Chapter 4.[38] Note that the approch presented in this chapter differs significantly from other recent approaches on discriminatively trained AF-based models [Tang et al., 2012; Jyothi et al., 2012] since our models are applied to a prediction task which also involves acoustics as opposed to the lexical access task studied in those works.

[38] A version of the work described in this chapter has appeared previously in [Prabhavalkar et al., 2013].

In this chapter:

- We conduct experiments to determine the feasibility of incorporating an articulatory feature-based pronunciation model for STD and evaluate the models in the setting of limited data by simulating sets of increasing size by sampling utterances from the Switchboard [Godfrey et al., 1992] dataset. In experimental results, we find that that systems with AF-based pronunciation models improve performance over phone-based models in some settings.

- We determine the impact of allowing for asynchronous feature transitions in our models. We find evidence that the models hypothesize greater asynchrony for those examples that likely contain larger amounts of pronunciation variation.

We begin in Section 5.2 by describing how the models developed in Chapter 4 can be adapted to incorporate an AF-based pronunciation model. As we mentioned briefly in Chapter 4, in the current work it is particularly advantageous that our models do not require knowledge of ground-truth AF alignments, since as we have seen in Chapter 3 these are hard to estimate directly from the acoustics. Instead, in the current work these are treated as latent variables in the model. We present the results of experiments conducted on the same sets as in Chapter 4, where we simulate the setting of limited training data, in Section 5.3. Additional evaluation and analysis of the models in presented in Section 5.3.2 to determine the impact of allowing articulatory asynchrony in the models. We evaluate the impact of allowing the model to hypothesize additional asynchrony in Section 5.4 and end with a brief summary of the chapter in Section 5.5.

5.1 Articulatory Feature-based Model: Notation and Preliminaries

We use the same notation as in Section 4, modified to represent articulatory feature streams as opposed to the phone-based representation used in that chapter. The articulatory feature-based pronunciation model used in this work is based on the model presented in Chapter 3. Pronunciations are modeled in terms of a set of articulatory feature streams based on the tract variables of articulatory phonology [Browman and Goldstein, 1992] which represent the configurations of the speech articulators: the constriction degrees and positions of the lips, the tongue tip, the tongue body, and the state of the velum and the glottis.[39]

We model pronunciation variation using gestural overlaps by allowing the articulatory streams to transition asynchronously from one target state to the next. When all AF streams are synchronized, the resulting surface pronunciation corresponds (by construction) to the canonical pronunciation; asynchronous transitions can model non-canonical pronunciations. In particular, such a model can account for non-canonical variant pronunciations with vowel nasalization, anticipatory/preservatory rounding, and epenthetic stop insertion. This is illustrated in Figure 5.1 which shows a non-canonical variant pronunciation of "sense".[40]

Formally, we model pronunciation via a set of K articulatory feature streams. We assume that the waveform is parameterized into acoustic feature vectors (e.g., PLPs) $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, where T is the number of frames in the utterance and where $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ is a feature vector for the t^{th} frame. Given an utterance $\overline{\mathbf{x}}$ and a query term $\overline{v}$, we denote

[39] As was the case with our experiments in Chapter 3, we assume that all features corresponding to the lips are completely synchronized as are the features corresponding to the tongue and the combination of glottis and velum. Thus, $K = 3$ in our experiments.

[40] As we have previously noted, our model does not explicitly model reduction in gestural magnitudes through AF substitution; this has been explored in other works [Livescu and Glass, 2004a,b; Jyothi et al., 2011]) but is not modeled explicitly here. However, this is implicitly modeled through the use of features derived from AF classifiers used as feature functions in our feature maps; see subsequent sections.

by $|\overline{v}|$ the number of phones in the canonical pronunciation of $\overline{v}$. In order to represent pronunciations in terms of articulatory feature streams, we assume that we have access to a phone-based pronunciation dictionary that maps each word v in the lexicon ($\mathcal{V}$) to its corresponding sequence of phone targets $\pi(v) \in \mathcal{P}^*$, where $\mathcal{P}$ represents the phone set. We then map the corresponding phone targets to corresponding articulatory feature targets, expanding from the mapping defined in [see Livescu, 2005, Appendix B] to ensure a unique AF configuration for each phone. We denote the corresponding sequence of articulatory targets for stream i as $(\sigma^i_1, \sigma^i_2, \cdots, \sigma^i_{|\overline{v}|})$. For a given hypothesized start and end time, $(1 \leq s < e \leq T)$, we denote a valid articulatory segmentation $\overline{\mathbf{s}}$ of $\overline{v}$ as the matrix of values that represent the start and end times for each of the AF states: $\overline{\mathbf{s}}_{i,j} = s^i_j$ where s^i_j is the start time of the j^{th} unit in stream i (i.e. σ^i_j). Thus, $s = s^i_1 < s^i_2 < \cdots < s^i_{|\overline{v}|} < e$, so that the state j in stream i extends from $t = s^i_j$ to $t = s^i_{j+1} - 1$, where $s^i_{|\overline{v}|+1} = e + 1$. We use the notation $\overline{\mathbf{s}} \sim (s, e)$ to denote an articulatory segmentation $\overline{\mathbf{s}}$ that begins at frame s and ends at frame e. In order to reduce computational complexity and eliminate implausible segmentations [Prabhavalkar et al., 2011], we restrict the amount of asynchrony to some number of states M: For all pairs of streams i, j and for each unit $1 \leq k \leq |\overline{v}|$ in the pronunciation, the extent of σ^i_k must lie between the extents of the succeeding and preceding M units in all other streams,

$$s^j_{k-M} \leq s^i_k \qquad \text{and} \qquad s^i_{k+1} \leq s^j_{k+M} \tag{5.1}$$

In particular, setting $M = 0$ would enforce complete synchrony. Finally, we denote the AF value for stream i hypothesized at time frame t under segmentation $\overline{\mathbf{s}}$ as $p^i_t(\overline{\mathbf{s}})$, i.e. $p^i_t(\overline{\mathbf{s}}) = \sigma^i_j$ for $s^i_j \leq t < s^i_{j+1}$. Our notation is presented in Fig. 5.1.

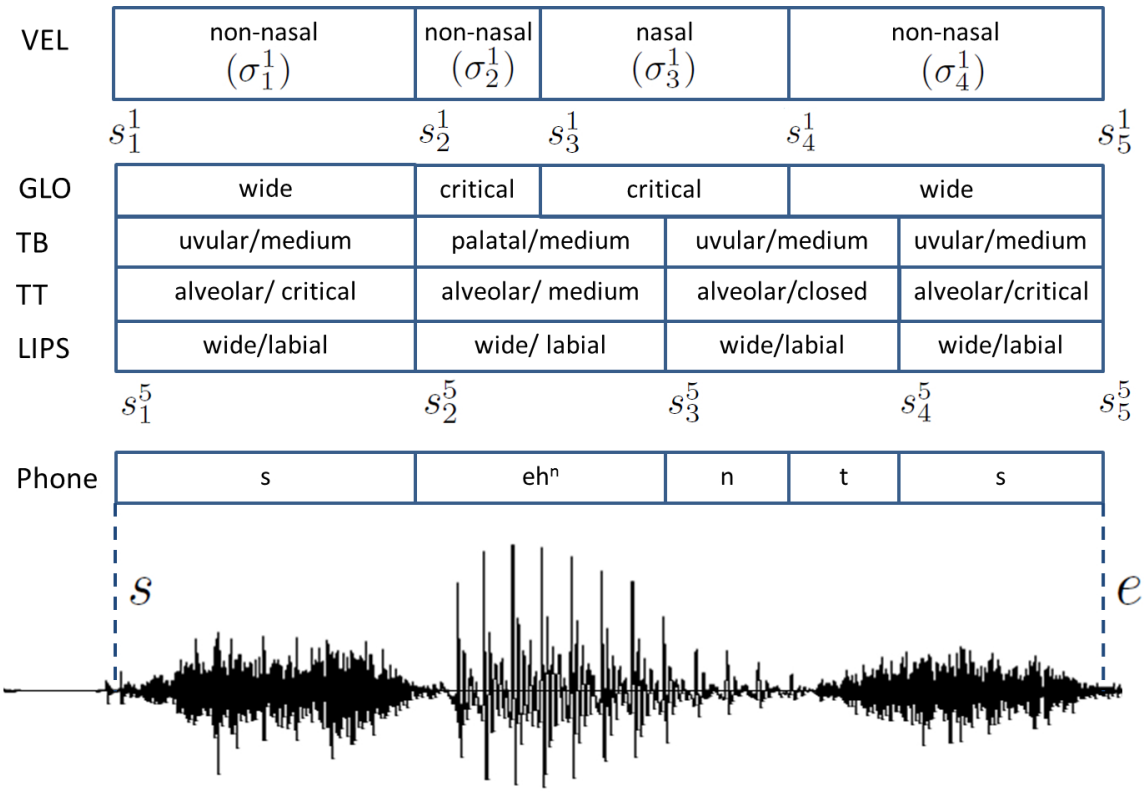

Figure 5.1: Non-canonical pronunciation of the word ‘sense’. The glottis and velum desynchronize from the other features, producing an epenthetic [t] and nasalized [eh].

5.2 Discriminative Model for STD

Our STD function is identical to the one presented in Section 4.3, except that the underlying pronunciation model is replaced with an articulatory feature based model; the only difference between the two models lies in the interpretation of the articulatory segmentation $\overline{\mathbf{s}}$ (now a matrix instead of the vector represented in Section 4.3) and in the form of the feature maps. Following [Keshet et al., 2009], our STD function is parameterized by a set of linear weights $\mathbf{w} \in \mathbb{R}^n$, as

$$f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{v}) = \max_{\overline{\mathbf{s}} \in \mathcal{S}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}) \tag{5.2}$$

where $\mathcal{S}$ is the set of all valid articulatory segmentations and $\boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}) \in \mathbb{R}^n$ is a feature vector. The score in Eq. 4.1 corresponds to the score of the highest scoring segmentation, $\overline{\mathbf{s}}$, over all possible start and end times within the utterance $\overline{\mathbf{x}}$ for the term $\overline{v}$. The feature vectors, $\boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}})$, are composed of a set of pre-defined feature maps $\{\phi_j\}_{j=1}^m$, where $\phi_j : \mathcal{X}^* \times \mathcal{V}^* \times \mathcal{S} \rightarrow \mathbb{R}^r$. Each feature map takes as input the acoustics $\overline{\mathbf{x}}$, the term $\overline{v}$, and the articulatory segmentation $\overline{\mathbf{s}}$ and returns an r-dimensional vector. As we noted previously, although the maximization in Eq. 4.1 is over $O(T^{|\overline{v}|K})$ possible segmentations, the maximizing segmentation can be computed using dynamic programming as described in [Prabhavalkar et al., 2011].

5.2.1 Feature Maps

We use two types of feature maps analogous to those used in our previous work on phone-based STD Prabhavalkar et al. [2012] presented in Chapter 4. Our feature maps are constructed from a set of *feature functions* $\boldsymbol{\xi} : \mathcal{X} \rightarrow \mathbb{R}^r$ computed from the acoustic frames, $\mathbf{x}$. The use of arbitrary feature functions allows us to leverage diverse sources of

information; as was the case in the previous chapter, in this work, we use PCA-transformed log posteriors of MLP classifiers of AFs and phones. We generically denote the extraction of the feature functions from acoustic vectors as $\boldsymbol{\xi} : \mathcal{X} \rightarrow \mathbb{R}^r$, which takes as its input an acoustic vector and returns a vector of feature values.

Given a suitable feature function $\boldsymbol{\xi}(\cdot)$, our first set of feature maps compute the confidence that the acoustic frames correspond to the hypothesized configurations of AFs:

$$\phi_{1,q^1,\cdots,q^K} = \frac{1}{e-s+1} \sum_{t=s}^{e} \boldsymbol{\xi}(\mathbf{x}_t) \delta[p_t^1(\bar{\mathbf{s}}) = q^1 \wedge \cdots \wedge p_t^K(\bar{\mathbf{s}}) = q^K] \tag{5.3}$$

where each $q^i \in \mathcal{Q}^i$ is a possible value that can be assigned to the i^{th} AF stream and $\delta[a] = 1$ if the condition a is true and 0 otherwise. Thus, we have $|\mathcal{Q}_1| \times \cdots \times |\mathcal{Q}_K|$ features maps of the first type, each of which is a vector of length equal to the length of $\boldsymbol{\xi}$.

The second set of feature maps correspond to AF state transitions, capturing the specific characteristics of the acoustics at AF transitions,

$$\phi_{2,i,q_1^i,q_2^i} = \frac{1}{e-s+1} \sum_{t=s+1}^{e} \boldsymbol{\xi}(\mathbf{x}_t) \delta[p_{t-1}^i = q_1^i \wedge p_t^i = q_2^i] \tag{5.4}$$

where $q_1^i, q_2^i \in \mathcal{Q}^i$ are possible states for stream i. As in Equation 5.3, each feature map is a vector of length equal to the length of $\boldsymbol{\xi}$ with a total of $\sum_{i=1}^{K} |\mathcal{Q}^i|^2$ feature maps of this type. As before, the feature maps in Equations 5.3 and 5.4 are normalized by the length of region in which the term has been hypothesized, in order to make scores comparable across different segment lengths. Also, note that if the model contains only a single stream ($K = 1$), whose values correspond to the phoneme sequence in the term's pronunciation, then the resulting feature maps are identical to those used in our previous STD approach using a phone-based model presented in the previous chapter (Section 4.3.1).

5.3 Experiments

Our experimental setup is identical to the setup in the experiments presented in Section 4.5 on subsets of Switchboard [Godfrey et al., 1992]: we evaluate performance obtained by training on four sets containing 500–5000 utterances drawn from Switchboard sets 23–49, parameters are tuned on the 40 term development set and results are reported on the test set containing 60 terms. For each term in the development and test sets, we consider 20 utterances containing the term (positive utterances) and 20 utterances that do not contain the term (negative utterances), drawn from Switchboard sets 20–22. Feature functions $\boldsymbol{\xi}(\mathbf{x})$ are modeled as 'tandem' feature projections onto the top 39 prinicipal components using PCA of the log-posteriors from four MLPs trained to predict L, T, G configurations, and phones, which serve as feature functions in our discriminative STD systems (after appending a constant bias term, so that $\boldsymbol{\xi}(\mathbf{x}) = 40$) and as acoustic features in our GMM-HMM baselines (monophone and triphone keyword-filler models, as described in Section 4.5). In order for our results to be comparable with the baselines, we model each AF label using 3-state models. The discriminative models are trained to optimize AUC using the algorithm described in Figure 4.5, suitably adapted to include articulatory segmentations, instead of the phone segmentations used in Chapter 4. More details can be found in Section 4.5.

Expanding the Phone to Articulatory Feature Mapping

As mentioned in Sec. 5.1, we determine the pronunciation of the word in terms of its articulatory feature representations by mapping phone-based pronunciations to their corresponding articulatory configurations using a deterministic mapping [see Livescu, 2005, Appendix B]. However, under this mapping, some phone configurations are mapped to the

same articulatory feature configurations (e.g., /r/ v.s. /er/). In pilot experiments, we observed that this had a detrimental effect on system performance in our articulatory feature-based systems. We therefore modified the mapping in [Livescu, 2005] by expanding the set of labels for the configurations of glottis-velum (G labels; adding 5 new labels) that ensured that every pair of phones that would have been mapped to the same L, T, G configuration under the mapping in [Livescu, 2005] now differed in the value of the G label. Thus, no two phones in our system are mapped to the same L, T, G configuration.

5.3.1 Results I: Incorporation of AF-based Pronunciation Model

We present results for each of the four training sets, which compare performance obtained using the monophone and triphone baselines (with acoustic models as tandem features), and the discriminative phone-based STD model (Chapter 4) against the AF-based discriminative systems allowing either one state of asynchrony (Disc-AF-1; $M = 1$) or no asynchrony (Disc-AF-0; $M = 0$), and assigning 3 states per AF label. Note that the system with no asynchrony is not identical to a discriminative phone-based system, because of the difference in the form of the feature maps modeling transitions. The results are summarized in Table 5.1. As can be seen in the table, all of the discriminative systems significantly outperform the monophone HMM baseline. For all training set sizes except 2500, the discriminative systems also outperform the context-dependent HMM baseline. This is particularly encouraging, because our discriminative systems are context-independent. Incorporating context-dependence is straightforward in our work, and we leave this for future work. The AF-based systems significantly outperform the phone-based discriminative system in the lowest-data case ($p < 0.01$). In the highest data case, the difference between Disc-AF-1 and Disc-Phone is at a significance level of $p = 0.033$. The AF-based system

System	500	1000	2500	5000
HMM-mono	0.810	0.827	0.846	0.857
HMM-tri	0.828	0.855	0.899	0.920
Disc-Phone	0.874*	0.901*	0.917	0.933*
Disc-AF-0	0.885*,†	0.897*	0.914	0.937*
Disc-AF-1	0.888*,†	0.898*	0.915	0.939*,†
Disc-Phone-AF-1	0.891*,†	0.905*	0.920*	0.940*,†

Table 5.1: AUC averaged over 60 query terms in the test set for systems trained on 500–5000 utterances. (*, †) represent significant ($p \leq 0.05$) improvements over HMM-tri and Disc-Phone, respectively, using a one-tailed Wilcoxon signed-ranks test. Performance of the discriminative systems relative to the monophone HMM system (HMM-mono) is strongly significant ($p \leq 0.001$) across all training set sizes.

with asynchrony (Disc-AF-1) slightly outperforms the synchronous system (Disc-AF-0) across data set sizes, but the differences are insignificant (the most significant difference between Disc-AF-0 and Disc-AF-1 is at a significance level of $p = 0.052$ in the lowest-data setting).

Since both phone- (Disc-Phone) and feature-based (Disc-AF-1) systems are themselves linear models, it is straightforward to combine them into a single linear model (Disc-Phone-AF-1):

$$f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{v}) = \max_{\overline{\mathbf{s}}_{\mathrm{P}}, \overline{\mathbf{s}}_{\mathrm{AF}}} \mathbf{w}_{\mathrm{P}} \cdot \boldsymbol{\phi}_{\mathrm{P}}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}_{\mathrm{P}}) + \mathbf{w}_{\mathrm{AF}} \cdot \boldsymbol{\phi}_{\mathrm{AF}}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}}_{\mathrm{AF}}) \tag{5.5}$$

where, we constrain $\overline{\mathbf{s}}_{\mathrm{P}}$ and $\overline{\mathbf{s}}_{\mathrm{AF}}$ to have the same start and end times. The weights $\mathbf{w}_{\mathrm{P}}$ and $\mathbf{w}_{\mathrm{AF}}$ are initialized using the fully trained Disc-Phone and Disc-AF-1 models respectively. We then train the entire model discriminatively. This system combination (Disc-Phone-AF-1) improves performance further, significantly outperforming HMM-tri in every case and the discriminative phone-based system in the lowest- and highest-data cases.

Figure 5.2: Fraction of hypothesized asynchronous states vs. "canonicalness" of the pronunciation, for the 100 query terms in the development and test sets in the 5000-utterance condition. Each point represents one of the 100 query terms in the development and test sets.

5.3.2 Analysis of Asynchronous AF-based System

Since we did not find a significant difference in performance betwen the synchronous ($M = 0$) and asynchronous ($M = 1$) AF-based STD systems, we further analyze the behavior of the Disc-AF-1 system to determine what impact, if any, asynchrony had on the system in terms of the situations in which the system hypothesizes asynchrony. We computed unconstrained phonetic decodings using the monophone HMM-based system trained on 5000 utterances (128 Gaussian components per mixture) on the portion of the positive utterances corresponding to the query term. The phonetic accuracies of these decodings against the canonical pronunciations give a rough measure of pronunciation variation in utterances of that term.[41] We then examined the segmentations hypothesized by the AF-based system to determine the percentage of states that are asynchronous. In Figure 5.2 we plot this percentage of asynchronous states versus the "canonicalness" measure for each keyword in the development and test sets. As we would expect, the plot seems to indicate that the AF-based system hypothesizes a greater amount of asynchrony for utterances with higher pronunciation variation. Note that the systems were not trained using any information of ground truth or estimated articulatory segmentation information. This provides some evidence for the fact that our systems are indeed modeling some of the pronunciation variation that arises from gestural overlaps.

5.4 Results II: Effect of Allowing Addtional Asynchrony in the Models

All of the AF-based pronunciation models presented thus far have allowed for up to one state of asynchrony ($M = 0$ or 1) in the model. The final experiment presented in

[41] Although factors besides pronunciation variation, for example background noise, might also be responsible for low phone recognition rates, phone recognition accuracies are likely to be highly correlated with pronunciation variation.

System	1000	2500	5000
Disc-AF-0	0.897	0.914	0.937
Disc-AF-1	0.898	0.915	0.939
Disc-AF-2	0.896	0.917	0.939

Table 5.2: AUC averaged over 60 query terms in the test set for systems trained on 1000–5000 utterances.The differences between the various systems are not significant ($p > 0.05$) using a one-tailed Wilcoxon signed-ranks test.

this chapter examines the effectiveness of allowing the model to hypothesize additional asynchrony by allowing up to two states of relative asynchrony between adjacent feature streams $M = 2$. Unfortunately, these models are significantly slower to train and evaluate than the models with $M = 0$ or 1 (cf., Section 3.5.3). The results of these experiments are reported in Table 5.2 when trained on sets of size 1000–5000 utterances. As can be seen in the table, there were no significant differences in the performance of the model as the amount of allowed asynchrony was increased.

There are two possible explanations of the results in Table 5.2. One possibility is that some of the variation is already being captured in the MLP feature detectors at the level of the 'acoustic model', thus effectively hiding it from the pronunciation model. The second possibility is that the lack of significant differences between the various systems is an artifact of our evaluation paradigm; our STD setup is designed to detect whether a small set of words can be accurately detected from a limited set of speech utterances. It is possible that the models that allow for additional relative asynchrony do indeed model the underlying pronunciation variability, but that this occurs rarely enough in our set that it does not impact overall performance.

System	Async. Frames in All Examples	Async. Frames in Positive Examples	Async. Frames in Negative Examples
Disc-AF-1	9.5%	4.8%	13.5%
Disc-AF-2	6.9%	3.0%	10.2%

Table 5.3: Analysis of asynchrony in the Disc-AF-1 and Disc-AF-2 systems. The table lists the fraction of frames corresponding to the maximizing articulatory segmentation that are asynchronous for (a.) all examples, (b.) positive examples, and (c.) negative examples, in the development and test set.

In order to investigate the second possibility further, for each utterance ($\overline{\mathbf{x}}$) and query term ($\overline{v}$) in the development and test sets, we examine the maximizing articulatory segmentation: $\overline{\mathbf{s}}^* = \text{argmax}_{\overline{\mathbf{s}} \in \mathcal{S}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{v}, \overline{\mathbf{s}})$. For each frame of this segmentation ($\overline{\mathbf{s}}^*$) we determine the percentage of frames which are asynchronous (at least one articulatory feature stream is de-synchronized from one of the other feature streams). In Table 5.3, we list percentage of asynchronous frames in (a.) all utterances in the development and test sets, (b.) positive utterances in the development and test sets, and (c.) negative examples in the development and test sets, for the AF-based systems trained on 5000 utterances.

Some interesting observations can be made based on the results in Table 5.3. First observe that the amount of asynchrony hypothesized in the negative examples is greater than in the positive examples. Intuitively, for negative utterances, the models are trying to find the best 'fit' for the query term in an utterance where the term does not exist; hypothesizing additional asynchrony provides the model with additional opportunities to find a good fit.[42] In other words, although all segmentations score poorly, more asynchronous segmentations might score higher on average. This might also explain why the percentage of asynchrony hypothesized in the Disc-AF-2 system is less than that in the Disc-AF-1 system: a model with additional flexibility ($M = 2$ vs. $M = 1$) provides more opportunities to find high

[42]Informally speaking, the model tries to make the best of a bad situation.

System Disc-AF-2	Async. Frames in All Examples	Async. Frames in Positive Examples	Async. Frames in Negative Examples
Relative asynchrony = 1	4.6%	2.2%	6.6%
Relative asynchrony = 2	2.3%	0.8%	3.6%

Table 5.4: Analysis of asynchrony in the Disc-AF-2 system trained on 5000 utterances in terms of how much relative asynchrony is hypothesized in the frames of the maximizing segmentation for terms in the development and test sets. The entries in the table corresponding to "Relative asynchrony = 1" indicates the fraction of frames for which the relative asynchrony between any pair of feature streams is only one state. Entries corresponding to "Relative asynchrony = 2", on the other hand, indicate the fraction of frames for which the the maximum allowed asynchrony of two states is hypothesized in the maximizing segmentations.

scoring segmentations than a more constrained model; during training, the system might therefore learn parameter settings that constrain the amount of hypothesized asynchrony, in general. In the case of the positive examples on the other hand, a well-trained model would score correct segmentations of the words more highly than incorrect segmentations; we may speculate, under this interpretation, that the values in the table for the positive examples might be representative of the amount of asynchrony in conversational speech.

Finally, in Table 5.4 we further analyze the Disc-AF-2 system trained on 5000 training utterances to determine what fraction of the asynchronous frames actually hypothesize the maximum allowed relative asynchrony of two states. The table lists the fraction of states in the maximizing segmentations that correspond to at most one state of relative asynchrony between any pair of states ("Relative asynchrony = 1") and the fraction of states where the maximum allowed asynchrony is hypothesized ("Relative asynchrony =2"). As can be seen, a very small fraction of positive examples actually hypothesize two states of asynchrony.

Although the analyses presented in this section offer interesting insights into the behavior of the asynchronous AF-based systems, they do not completely explain why the asynchronous AF-based systems did not outperform the synchronous AF-based systems unlike in previous work on lexical access tasks [Livescu, 2005]. Performance might be improved further by the addition of a more detailed model of articulatory asynchrony; such a model can be incorporated into our discriminative framework through the use of more elaborate feature maps, thus allowing for a more constrained model of asynchrony. For example, this would allow us to model the dependence of asynchrony on factors such as position within the word, speaking rate, unigram language model probability of the word, or other factors which are known to correlate with pronunciation variation.

5.5 Summary

In this chapter we presented results on spoken term detection in the setting of limited training data using a discriminative articulatory feature-based pronunciation model. In experimental results, we found that the proposed system outperformed baseline HMM-based systems across a range of training set sizes and the discriminative phone-based STD systems in some settings. By analyzing the asynchronous AF-based pronunciation model, we observed that that the system appears to hypothesize a greater amount of asynchrony for examples which seem to contain more pronunciation variation, although we do not see significant differences in performance when the model is allowed to hypothesize additional asynchrony between AF streams.

CHAPTER 6: LEVERAGING EXISTING LVCSR-BASED SPOKEN TERM DETECTION SYSTEMS FOR DISCRIMINATIVE SPOKEN TERM DETECTION

The experiments on spoken term detection (STD) presented in the last two chapters demonstrated the effectiveness of the proposed discriminative STD system in low-resource conditions in experiments on a subset of the Switchboard database of conversational telephone speech [Greenberg et al., 1996]. One of the significant drawbacks of the approach presented in Chapters 4 and 5 is that the model must be re-evaluated for each term of interest. This is a direct consequence of the fact that the spoken term detection function, $f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{v})$, is computed explicitly with respect to a given term of interest. Thus, the complexity of evaluating a new query term scales linearly with the size of the test corpus, which might be prohibitively slow for large datasets. We end the thesis by describing how our techniques for STD can be adapted in situations where existing large vocabulary continuous speech recognizers (LVCSR) are available.[43]

[43]The research described in this chapter was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

The experiments in this chapter are performed on data from the IARPA Babel Program Cantonese language collection release babel101b-v0.4c [IAR, 2011]. The baseline system and the STD index used in our experiments were built by the Swordfish team – a joint collaboration between the International Computer Science Institute (ICSI), Columbia University, Northwestern University, The Ohio State University, and the University of Washington – for the BABEL evaluation. In particular, we acknowledge the contribution of Steven Wegmann, Arlo Faria and Adam Janin at ICSI for help in setting up the baseline STD system. We also thank Van Hai Do (Nanyang Technological University, Singapore) and Joo-Kyung Kim (The Ohio State University) for the *toneme posterior* features and the *bottleneck features* used in our experiments. Finally,

Another aspect of the discriminative STD approach that we investigate in this chapter is whether the process of training pair selection (i.e., selecting pairs of positive and negative utterances, that do and do not contain specific terms of interest) has an impact on the effectiveness of the learned STD system. In particular, we investigate whether performance can be improved by specifically selecting utterances that are incorrectly hypothesized by an LVCSR system to contain particular terms as negative examples for that term.

Further, we examine whether the performance of the discriminative (acoustic-only) STD techniques presented in previous chapters can be improved further by leveraging existing LVCSR-based STD systems. We end the chapter, by reporting the results of pilot STD experiments using articulatory feature-based (AF-based) pronunciation models for Cantonese.

In Section 6.1, we begin with a description of the dominant STD paradigm based on the use of trained LVCSR systems. We provide details of the baseline system used in this chapter, which is applied to the task of STD on conversational Cantonese telephone speech, in Section 6.3. We describe the experimental setup used in our experiment in Section 6.4 and describe our experimental results with phone-based discriminative models in Section 6.5 and AF-based discriminative models in Section 6.7. In Section 6.8 we compare the results of our discriminative spoken term detection experiments in English and Cantonese. In Sections 6.9 and 6.10 we describe how the proposed techniques for AUC optimization can be adapted in order to optimize averaged term weighted value (ATWV) [Fiscus et al., 2007], which is the evaluation metric used in the IARPA Babel evaluation [IAR, 2011]. We conclude with a summary of the work presented in this chapter in Section 6.11.

some of the work presented in this chapter developed out of a collaboration with Yanzhang (Ryan) He; his assistance, particularly in developing the scoring scripts, is gratefully acknowledged.

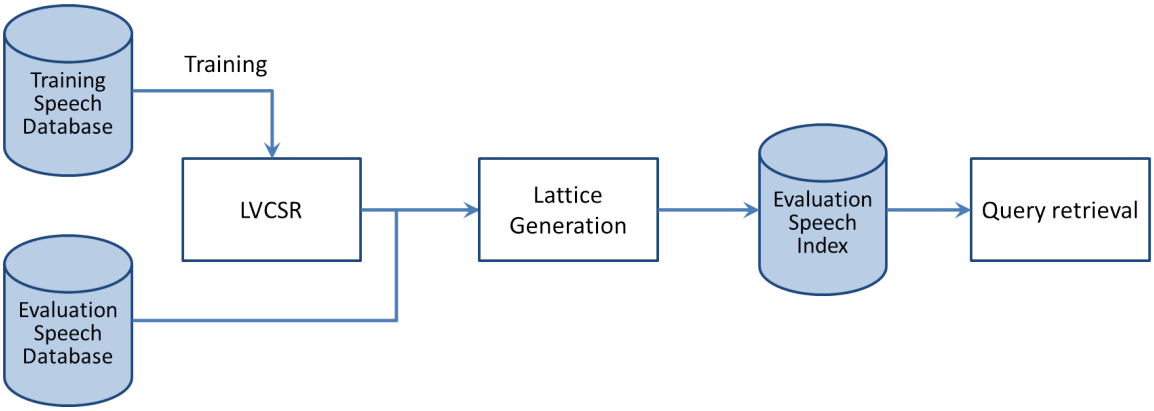

Figure 6.1: A schematic representation of the dominant paradigm in spoken term detection (STD) ([Miller et al., 2007; Vergyri et al., 2007; Akbacak et al., 2008] inter alia.). A baseline LVCSR system is first trained using a corpus of training data. The trained LVCSR system is then used to generated word lattices for the evaluation data. These lattices are then converted into a data structure known as the *index* that is used for subsequent processing. Detecting query terms is accomplished by searching the index to find instances of the respective terms. The advantage of this approach is that the evaluation speech database does not need to be re-processed in order to detect evaluation query terms.

6.1 LVCSR-based STD systems

The dominant paradigm for STD ([Vergyri et al., 2007; Miller et al., 2007; Akbacak et al., 2008] inter alia) relies on the availability of trained LVCSR systems to generate *speech indices*: a compact representation of the set of words hypothesized to be present in the speech utterances, along with a score that can be used to rank these hypotheses. Thus, the problem of STD can be reduced to the problem of searching and retrieval from the speech index. In the following sections, we discuss how the availability of such speech indices can be leveraged in our discriminative STD systems. A schematic representation of the LVCSR-based STD process appears in Figure 6.1. A detailed description of each of the steps in Figure 6.1 is deferred until Section 6.3.

Formally, we define a speech index, $\mathcal{I}$, as a set of five-tuples, $\mathcal{I} \subseteq \mathcal{X}^* \times \mathcal{V} \times \mathbb{N} \times \mathbb{N} \times [0,1]$, where $\mathcal{X}^*$ is the set of valid speech utterances and $\mathcal{V}$ is the lexicon of words,

$$\mathcal{I} = \{\langle \overline{\mathbf{x}}_i, v_i, s_i, e_i, P(v_i|\overline{\mathbf{x}}_i)\rangle\}_{i=1}^{N} \tag{6.1}$$

The elements of the five-tuples, $\langle \overline{\mathbf{x}}_i, v_i, s_i, e_i, P(v_i|\overline{\mathbf{x}}_i)\rangle$, represent respectively, the speech utterance $\overline{\mathbf{x}}_i \in \mathcal{X}^*$, the word $v_i \in \mathcal{V}$ hypothesized to be present in the utterance by the LVCSR system, the start and end times (s_i, e_i) within the utterance where the word is hypothesized to be present, and the posterior probability, $P(v_i|\overline{\mathbf{x}}_i)$, of the word in the utterance which represents the system's 'confidence' that the word was uttered in the given position in the utterance.[44]

6.2 Leveraging Speech Index for Discriminative STD

As we have discussed previously, the main limitation of the proposed discriminative STD approach, presented in Chapters 4 and 5 is that evaluating a dataset for the presence or absence of a particular speech term, $\overline{v} \in \mathcal{V}^*$, requires us to re-evaluate every speech utterance in the dataset for that particular term, $f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{v})$. Although this process can be trivially parallelized since scores assigned to specific utterances are independent of each other, the process may still be prohibitively slow for evaluating large speech datasets. The approach we take in this chapter is to use the available speech index to speed-up the process by effectively *re-scoring* the entries in the speech index. It should be stressed that this re-scoring process is carried out once for the entire index. Once this has been completed, the system can be evaluated to detect arbitrary query terms across the entire speech database.

[44]Each entry in the index corresponds to a particular arc in a decoded LVCSR lattice for that utterance. The posterior probability, $P(v|\overline{\mathbf{x}})$, for that arc is computed by accumulating the posterior probability of all paths in the lattice that pass through that arc.

6.2.1 Re-scoring the Speech Index

Assume that we have access to a trained discriminative STD system, $f_{\mathbf{w}}(\overline{\mathbf{x}}, v)$, parametrized by the weights $\mathbf{w}$ in the model as described in the previous chapters, and a speech index $\mathcal{I}$. We construct a new speech index $\mathcal{I}^{\text{disc}}$ from the original index $\mathcal{I}$ as follows,

$$\mathcal{I}^{\text{disc}} = \{\langle \overline{\mathbf{x}}_i, v_i, s_i, e_i, f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i, e_i, v_i)\rangle\}_{i=1}^{N} \tag{6.2}$$

Note that the term $f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i, e_i, v_i)$ appearing in Equation 6.2 corresponds to the score from our discriminative STD systems presented in Chapters 4 and 5,

$$f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i, e_i, v_i) = \max_{\overline{\mathbf{s}} \sim (s_i, e_i)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i, v_i, \overline{\mathbf{s}}) \tag{6.3}$$

Computing the re-scored speech index, $\mathcal{I}^{\text{disc}}$, is extremely fast and efficient: in Equation 6.3, we only consider (articulatory or phone) segmentations that begin and at at the position where the particular word v_i was hypothesized in the utterance.

Evaluating the index to detect a particular query term $\overline{v} = (v_1, v_2, \cdots, v_M)$, where each $v_j \in \mathcal{V}$, proceeds as follows:

1. Identify for each of the constituent words, v_j ($1 \leq j \leq M$), in the query term, all candidate entries in the index that correspond to that particular term,

$$\text{Cand}(v_j) = \left\{\left\langle \overline{\mathbf{x}}_i^j, v_j, s_i^j, e_i^j, f_{\mathbf{w}}(\overline{\mathbf{x}}_i^j, s_i^j, e_i^j, v_j)\right\rangle\right\}_{i=1}^{N(v_j)} \tag{6.4}$$

2. Given the set of constructed candidate sets, $\text{Cand}(v_j)$, identify entries across the candidate sets, which occur within a time-tolerance, Δ, of each other.[45] In other words, we construct the joint candidate set, $\text{Cand}(\overline{v})$:

$$\text{Cand}(\overline{v}) = \left\{\left\langle \overline{\mathbf{x}}_i, \overline{v}, s_i^1, e_i^M, f_{\mathbf{w}}{}^{\text{joint}}(\overline{\mathbf{x}}_i, \overline{v})\right\rangle\right\}_{i=1}^{N(\overline{v})} \tag{6.5}$$

[45] In accordance with the BABEL IARPA specifications, we set Δ to correspond to 0.5 seconds in these experiments.

where, $\left\langle \overline{\mathbf{x}}_i, \overline{v}, s_i^1, e_i^M, f_{\mathbf{w}}{}^{\text{joint}}(\overline{\mathbf{x}}_i, \overline{v}) \right\rangle \in \text{Cand}(\overline{v})$, if and only if,

$$\left\langle \overline{\mathbf{x}}_i, v_j, s_i^j, e_i^j, f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i^j, e_i^j, v_j) \right\rangle \in \text{Cand}(v_j) \tag{6.6}$$

$$\left| \frac{(s_i^j + e_i^j)}{2} - \frac{(s_i^{j+1} + e_i^{j+1})}{2} \right| \leq \Delta \quad \text{for all, } (1 \leq j \leq M-1) \tag{6.7}$$

$$f_{\mathbf{w}}{}^{\text{joint}}(\overline{\mathbf{x}}_i, \overline{v}) = \frac{1}{\sum_{j=1}^{M}(e_i^j - s_i^j + 1)} \sum_{j=1}^{M}(e_i^j - s_i^j + 1) f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i^j, e_i^j, v_j) \tag{6.8}$$

In other words, the conditions expressed in Equations 6.6–6.8 can be re-stated as requiring that each of the individual words in the the query term, $\overline{v}$, be present in the same utterance (Equation 6.6) and that the individual terms must be located in close proximity (Equation 6.7) and that the *joint* score for the query term is the sum of individual candidate scores weighted by the number of frames, in order to be consistent with the fact that the discriminative system scores $f_{\mathbf{w}}(\overline{\mathbf{x}}_i, s_i^j, e_i^j, v_j)$ are individually normalized by their respective lengths (Equation 6.8).

3. The entries in the candidate set, Cand$(\overline{v})$, are then returned by the system as putative hits of $\overline{v}$ detected in the speech database if they score higher than a user-defined threshold. As per the Babel program specifications, the particular entries are judged to be correct detections if the mid-point of the detected term, $\frac{(s_i^1 + e_i^M)}{2}$, is within a tolerance of $(+/-)\Delta$ frames from the mid-point of a true occurence of the term, and are otherwise declared to be a false alarms.

6.3 Description of the Baseline System

Our experiments are conducted using data from the IARPA Babel Program Cantonese language collection release babel101b-v0.4c. The LVCSR system used in our experiments is a discriminatively trained (Minimum Phone Error [Povey and Woodland, 2002]) cross-word triphone system with speaker-adaptation, which represents a *strong* baseline that we

compare performance against.[46] We describe the baseline system in detail in the following sections, by elaborating on each of the steps that were indicated in Figure 6.1. The training of the baseline system proceeds in two stages, which are described separately in Section 6.3.1 and 6.3.2, respectively.

6.3.1 LVCSR Training: First Stage

In the first stage, "base" features are extracted from the training utterances which consist of 12th order MFCC coefficients with energy. The base features are warped with speaker-dependent Vocal Tract Length Normalization (VTLN) [Lee and Rose, 1998] warp factors and then mean and variance normalized. The base features are used to train a context-independent monophone HMM system using the standard HTK recipe [Young et al., 2002]. Phones are modeled as 3-state left-to-right HMMs, and output distributions are modeled as mixtures of Gaussians with diagonal covariance. This system, after training, is used to force-align the training data, to obtain monophone target labels for each frame of speech in the training set.

Pitch Feature Extraction

Pitch features are extracted from the training utterances using the Subband AutoCorrelation Classification (SAcC) pitch tracker [Lee and Ellis, 2012] and processed through a multilayer perceptron (MLP) to predict quantized pitches in the range 60 to 400Hz (plus a "no voice" class). After smoothing the MLP outputs, two pitch features are generated: (a.) the *log* of the pitch and (b.) the *log* probability of voicing for each of the speech utterances.

[46] As we mentioned in the introduction to this chapter, the baseline system was developed by the Swordfish team for the IARPA Babel evaluation [IAR, 2011]; their help and support is gratefully acknowledged.

Bottleneck Feature Training

Bottleneck features [Grézl et al., 2007] are computed using a hierarchical network consisting of two MLPs as follows. The "base" MFCCs and the two pitch features from the current frame are concatenated together with the features from the preceding and succeeding 7 frames (15-frames context), followed by the application of a discrete cosine transform (DCT) to obtain 16 coefficients at each frame. These 240 coefficients form the inputs to the first, seven-layer MLP which is used to generate 60-dimensional bottleneck features. This MLP is pre-trained layer-by-layer using restricted Boltzmann Machines (RBMs) [Hinton et al., 2006].

The 60-dimensional bottleneck features from the first MLP are used as inputs to the second, five-layer MLP, after concatenating them with frames that are at positions -10, -5, +5, +10 from the current frame to obtain a 300-dimensional input layer representation. The weights of the second MLP are randomly initialized following which the MLP is trained to predict the monophone targets generated as part of the first-stage HMM training. This second MLP is used to produce 30-dimensional bottleneck features corresponding to each frame.

The "base" 13-dimensional MFCCs and pitch features, together with their deltas, and double-deltas are appended together with the 30-dimensional bottleneck features (from the second MLP) to produce a 75 dimensional feature vector that is used for subsequent second-stage HMM training ("final" features).

Speech/ Non-speech Detector

An MLP is trained to predict whether a given frame in the speech waveforms corresponds to speech or not. The MLP targets are obtained as part of the first-stage HMM

training. This detector is used to perform a segmentation of the speech utterances into speech and non-speech regions (with corresponding word and phone transcriptions). These re-segmented speech waveforms are used for the second-stage HMM training.

6.3.2 LVCSR Training: Second Stage

The second stage HMM system is trained on the "final features" after re-segmenting the training data into speech and non-speech regions. This system is a conventional cross-word triphone system built using HTK [Young et al., 2002]. The system uses the standard HTK recipe: a monophone system is initially trained and cloned to form initial triphone models; triphone states are then clustered using phonetic decision trees to yield tied-state triphones. Observations are modeled as mixtures of Gaussians (diagonal covariance), with 16 components per mixture. The system is then discriminatively trained using the MPE criterion [Povey and Woodland, 2002]. In order to improve the modeling of covariance, a global semi-tied covariance (STC) transform [Gales, 1998] is applied to these models. The second-stage HMM system is used for subsequent processing steps of lattice and index generation.

6.3.3 Lattice Generation

A first-pass decoding from the second-stage HMM system is performed, using a trigram Kneser-Ney smoothed language model trained on the data provided as part of the BABEL language pack, to generate one-best hypotheses from the system. Next, speaker adaptation is performed using the one-best hypotheses by estimating maximum likelihood linear regression (MLLR) transforms [Leggetter and Woodland, 1995] on the data. Second-pass lattices are then generated for all of the data after speaker-adaptation. These lattices are used to generate the index.

6.3.4 Index Generation

The speech index, $\mathcal{I}$, is computed from the second-pass lattices. For a given utterance, $\overline{\mathbf{x}}$, every lattice arc is included as an entry $(\langle \overline{\mathbf{x}}, v, s, e, P(v|\overline{\mathbf{x}}) \rangle)$ in the index: the start and end times (s, e), and the hypothesized word, v, correspond to the lattice arc. The posterior probability, $P(v|\overline{\mathbf{x}})$, is computed by summing together the posterior probability of all paths in the lattice that pass through that arc. Multiple entries, with the same start-time, end-time and hypothesized word are consolidated together and represented as a single entry in the index whose posterior score is obtained by adding together the individual posterior scores.

6.3.5 Query Term Detection in Baseline System using the Index

Given a query term, $\overline{v} = (v_1, v_2, \cdots, v_M)$, the process of retrieval of the term is similar to the process described in Section 6.2.1 and is summarized briefly below:

1. Identify for each of the constituent words, v_j $(1 \leq j \leq M)$, in the query term, all candidate entries in the index, $\mathcal{I}$, that correspond to that particular term,

$$\text{Cand}(v_j) = \left\{ \left\langle \overline{\mathbf{x}}_i^j, v_j, s_i^j, e_i^j, P(v_j|\overline{\mathbf{x}}_i^j) \right\rangle \right\}_{i=1}^{N(v_j)} \tag{6.9}$$

2. Given the set of constructed candidate sets, $\text{Cand}(v_j)$, identify entries across the candidate sets, which occur within a time-tolerance, Δ, of each other. In other words, construct the joint candidate set, $\text{Cand}(\overline{v})$:

$$\text{Cand}(\overline{v}) = \left\{ \left\langle \overline{\mathbf{x}}_i, \overline{v}, s_i^1, e_i^M, P(\overline{v}|\overline{\mathbf{x}}_i) \right\rangle \right\}_{i=1}^{N(\overline{v})} \tag{6.10}$$

Set	Size	Notes
Training	6.5 hr	Gender balanced (Generated by Adam Janin at ICSI)
Development	1.65 hr	Demographically balanced (IBM conversational_heldout set)
Evaluation	1.85 hr	Gender balanced (Generated by Adam Janin at ICSI)

Table 6.1: Details of the training, development and evaluation sets used in the experiments described in this section. All sets are extracted from the babel101b-v0.4c data [IAR, 2011].

where, $\left\langle \overline{\mathbf{x}}_i, \overline{v}, s_i^1, e_i^M, P(\overline{v}|\overline{\mathbf{x}}_i) \right\rangle \in \text{Cand}(\overline{v})$, if and only if,

$$\left\langle \overline{\mathbf{x}}_i, v_j, s_i^j, e_i^j, P(v_j|\overline{\mathbf{x}}_i) \right\rangle \in \text{Cand}(v_j) \quad \text{for all, } (1 \leq j \leq M) \tag{6.11}$$

$$\left| \frac{(s_i^j + e_i^j)}{2} - \frac{(s_i^{j+1} + e_i^{j+1})}{2} \right| \leq \Delta \quad \text{for all, } (1 \leq j \leq M-1) \tag{6.12}$$

$$P(\overline{v}|\overline{\mathbf{x}}_i) = \min_j P(v_j|\overline{\mathbf{x}}_i) \tag{6.13}$$

It should be noted that the computation of the posterior for the entire query term, $P(\overline{v}|\overline{\mathbf{x}}_i)$, is approximated as the minimum of the individual posteriors for each of the constituent terms, $P(v_j|\overline{\mathbf{x}}_i)$. This is an approximation [Miller et al., 2007] necessitated by the fact that the lattices are discarded after index generation.

6.4 Experimental Setup

In order to determine the effectiveness of the proposed discriminative systems we conduct experiments on the IARPA Babel Program Cantonese language collection release babel101b-v0.4c. Since the baseline GMM-HMM system is trained on the 'training' portion of this data (approximately 50hrs of speech data), we create a separate (disjoint) training, development and evaluation sets using the babel101b-v0.4c_dev data (the 'full' BABEL development set). We refer to these three sets as train, development and evaluation respectively in subsequent sections (not to be confused with the 'original' BABEL training, development and evaluation sets). Details of these three sets appear in Table 6.1.

We use a set of $\sim$300 terms (development terms; chosen by the Babelon team at Raytheon BBN technologies) and 1000 terms (evaluation terms) which are used for evaluating STD performance on the development and evaluation sets respectively. Since the evaluation is restricted to a subset of the original BABEL development data, not all of the development and evaluation query terms occur in our sets. Only 115 of the 300 development terms and 125 of the 1000 evaluation terms occur in our chosen development and evaluation sets. Since both the baseline and the discriminative systems only score examples that appear in the baseline speech index, we ignore true occurences of the terms that do not appear in the index since these cannot be detected by either the baseline or by our discriminative systems.

6.4.1 Training Discriminative Systems: Leveraging the Index

All discriminative systems are trained on the data in the 6.5 hour training set. Since the discrimnative systems require the creation of pairs of positive and negative training instances, we consider two possible ways of generating these sets. In both cases, positive examples are extracted using the time-aligned word transcripts provided with the data. We extract each word that appears in the training data as a positive example for that word (60473 training examples). Negative examples corresponding to each positive examples are extracted using two different methodologies to examine their impact on system performance: in particular, whether performance can be improved by selecting challenging negative examples present in the index, $\mathcal{I}$:

- **Methodology I**: As in the experiments presented in Chapters 4 and 5, we randomly select an utterance that does not contain the word in the positive example as a corresponding negative examples.

- **Methodology II (impNegSel)**: Instead of randomly selecting negative examples from the set of all utterances that do not contain a given word, we restrict the candidate set of negative examples to those utterances in the index where the word is incorrectly hypothesized to exist (i.e., false alarms in the index). Intuitively, these examples represent challenging utterances that are likely to be acoustically confusable with the words in the query term. We refer to the systems trained using this methodology with the descriptor impNegSel (improved negative selection) in the results.

Feature Maps and Feature Functions in Discriminative Systems

The feature maps used in the discriminative phone- and feature-based systems are exactly the same as those described in Sections 4.3.1 and 5.2.1 which aim to capture local-frame level dependencies between the hypothesized sub-word labels and dependencies across adjacent frames. We compare performance obtained using two kinds of feature functions $\boldsymbol{\xi}(\mathbf{x})$: (a.) log posteriors of toneme classes (TonemeLogPost)[47] estimated using a deep neural network to which we append a bias term (i.e. $|\boldsymbol{\xi}(\mathbf{x})| = 131$), and (b.) bottleneck features computed during the baseline system training (Bottleneck), appended with a bias term ($|\boldsymbol{\xi}(\mathbf{x})| = 31$).

6.5 Results: AUC Performance from Discriminative Phone-based System

We report results in terms of AUC averaged across all instances of query terms on the development and evaluation sets from the baseline and discriminative systems in Table 6.2.

[47]The vowels in Cantonese are annotated with one of six tones. There are a total of 130 toneme classes in the dataset.

System	Average AUC Development Set	Average AUC Evaluation Set
HMM-avgACscore	0.616	0.626
HMM-post	0.821	0.844
Disc-TonemeLogPost	0.773*	0.810*
Disc-Bottleneck	0.790	0.792*
Disc-TonemeLogPost-impNegSel	0.794	0.814*
Disc-Bottleneck-impNegSel	0.798	0.807*

Table 6.2: Results of cantonese STD experiments obtained on the development and evaluation sets, reported in terms of averaged AUC across all terms in the respective sets. (*) indicates a statistically significant difference ($p \leq 0.05$) using a one-tailed Wilcoxon signed-ranks test relative to the HMM-post system. There is no significant difference between the performance of the system with or without improved negative example selection (impNegSel).

Since the discriminative systems are effectively 'acoustic-only' systems which do not utilize word-level context information such as language model scores, we also report results obtained using an 'acoustic-only' baseline HMM system. Thus, our two baselines are: (a.) an acoustic-only baseline (HMM-avgACscore) that uses the averaged (over frames) acoustic-model score for each entry in the index corresponding to the best segmentation of the pronunciation of the term: $\frac{1}{e-s+1} \max_{\overline{\mathbf{s}}} p(\mathbf{x}|\overline{\mathbf{s}}, \pi(\overline{v}))$ (b.) the baseline posterior score (HMM-post): $P(\overline{v}|\overline{\mathbf{x}})$.

As can be seen in the Table, comparing the discriminative phone-based system to the *acoustic-only* baseline, the discriminative systems outperform the baseline by large margins. The stronger HMM baseline, HMM-post, which also has access to word-level context information in the lattices scores about 2% absolute higher than the discriminative systems on the development sets and about 4-5% better on the evaluation set. This strongly suggests that word-level context information is important for good STD performance. Furthermore, the baseline HMM systems are trained on ∼50 hours of speech data, which is an order of

magnitude more data than was used to train the discriminative sysetms. In Section 6.6, we examine the effect of incorporating this additional context into the discriminative models where we find that interpolating baseline scores with the score from the discriminative systems results in large gains.

Another observation that can be made when examining the impact of the improved selection of negative examples (impNegSel) is that performance improves slightly, but the magnitude of the improvement is not consistently high on both the development and evaluation sets. The absolute improvement in the Disc-TonemeLogPost system ranges from 2.1% on the development set to 0.4% on the evaluation set, whereas the improvement for the Disc-Bottleneck system ranges from 0.8% on the development set to 1.5% on the evaluation set.

6.6 Results: Interpolating Discriminative System Scores with Baseline Posterior Scores

In order to determine whether the scores obtained using the discriminative baseline systems are complementary with respect to the baseline posterior system (HMMpost) we conduct experiments to determine whether performance can be improved further by linear interpolation of the scores from the two systems.

Given the scores ${f_{\mathbf{w}}}^{\text{joint}}(\overline{\mathbf{x}}, \overline{v})$ (Equation 6.8) and $P(\overline{v}|\overline{\mathbf{x}})$ from the discriminative system and the baseline HMM system for a particular occurrence of a search term, $\overline{v}$, we define the combined interpolated score for that instance as:

$$ {f_{\mathbf{w}}}^{\text{comb}}(\overline{\mathbf{x}}, \overline{v}) = {f_{\mathbf{w}}}^{\text{joint}}(\overline{\mathbf{x}}, \overline{v}) + \mu P(\overline{v}|\overline{\mathbf{x}}) \tag{6.14} $$

where, $\mu \in [0, \infty)$ represents the interpolation weight.

We only report results of the interpolation process (in terms of averaged AUC) for the discriminative system trained on the bottleneck features, since the systems employing both toneme log-posteriors and bottleneck features performed similarly (cf., Table 6.2). In Figure 6.2, we plot performance obtained on the development set as a function of the interpolation weight. As can be seen in the figure, the performance improves significantly after linear interpolation of scores, increasingly steadily as the interpolation weight increases. Perhaps what is more suprising is the magnitude of the improvement on the development set: 7.0% for the Disc-Bottleneck system and 8.4% for the Disc-Bottleneck-impNegSel system. One explanation of the fact that the system with improved negative selection improves performance more after interpolation with the baseline might be due to the fact that the examples that the system learns to 'separate' are precisely those examples where the baseline system incorrectly hypothesizes a given word to be present. The results on the development and evaluation sets are summarized in Table 6.3. Comparing the performance of the two interpolated systems (Interp-HMMpost/Bottleneck and Interp-HMMpost/Bottleneck-impNegSel), although there was a very significant difference ($p = 0.009$) between the two interpolated systems on the development set, there was no significant difference between them on the evaluation set ($p = 0.139$). Individually, however, each system significantly improved performance over both the baseline system and the individual discriminative systems trained on Bottleneck features. Overall, the interpolated systems improve performance over the baseline by between 3.4–4.4%.

6.7 Pilot Experiment: Incorporating AF-based Pronunciation Models

In order to determine whether incorporating an articulatory feature-based pronunciation model can improve performance on the Cantonese data as well, we conduct experiments on

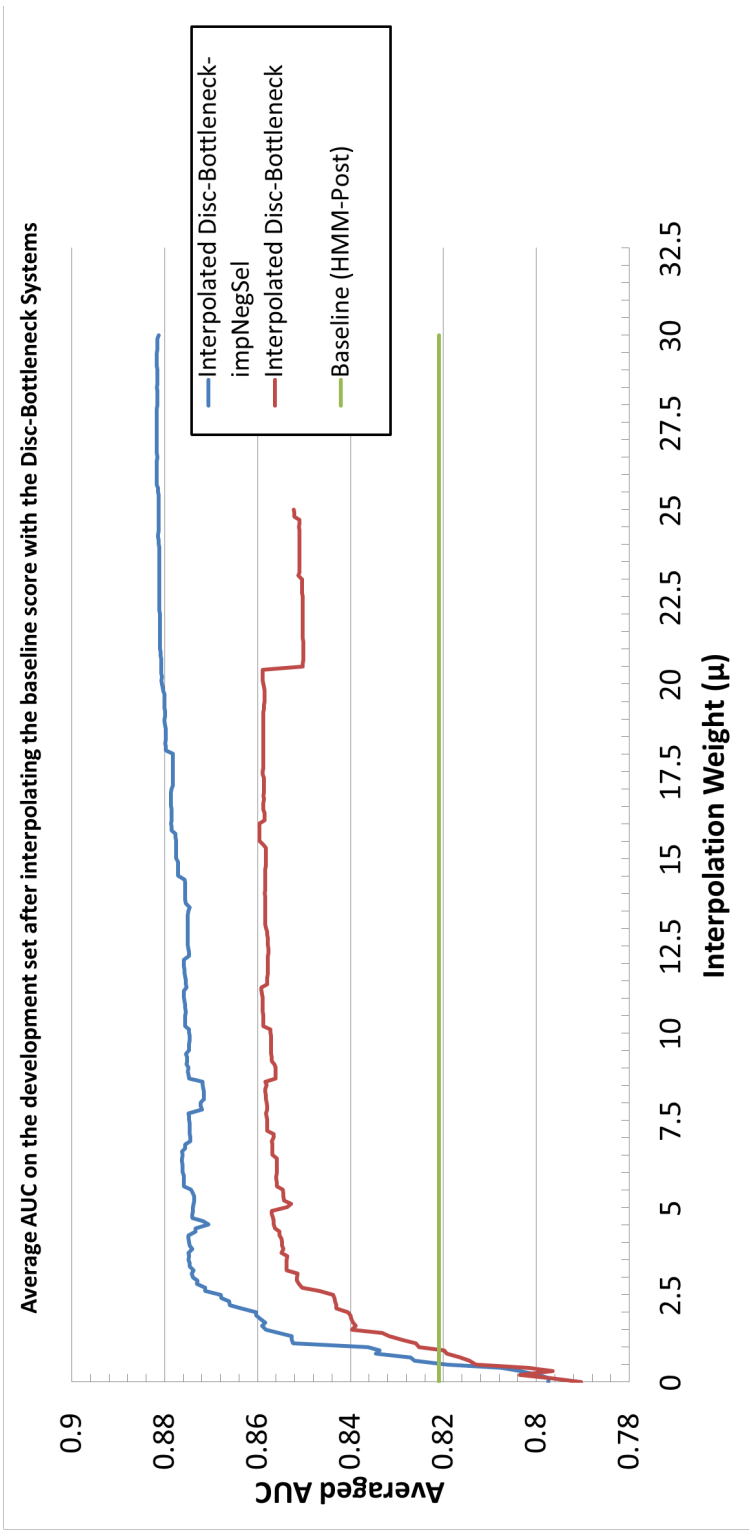

Figure 6.2: Performance in terms of averaged AUC obtained by interpolating the baseline system (HMMpost) with the discriminative systems trained on bottleneck features (Disc-Bottlenecnect-impNegSel) and (Disc-Bottleneck).

System	Avg. AUC Dev. Set	Avg. AUC Eval. Set
HMM-avgACscore	0.616	0.626
HMM-post	0.821	$0.844^{\dagger}$
Disc-Bottleneck	0.790	0.792
Disc-Bottleneck-impNegSel	0.798	0.807
Interp-HMMpost/Bottleneck ($\mu = 15.5$)	$0.860^{*,\dagger}$	$0.878^{*,\dagger}$
Interp-HMMpost/Bottleneck-impNegSel ($\mu = 29.1$)	$0.882^{*,\dagger}$	$0.888^{*,\dagger}$

Table 6.3: Results of Cantonese STD experiments obtained on the development and evaluation sets, reported in terms of averaged AUC. (*) indicates a statistically significant improvement ($p \leq 0.05$) using a one-tailed Wilcoxon signed-ranks test over the HMM-post system. (†) represents a statistically significant improvement ($p \leq 0.05$) over either of the Disc-Bottleneck systems.

the system that uses the bottleneck features as our feature functions. We constructed a mapping from Cantonese phoneme categories to corresponding L (6 values), T (16 values) and G (30 values) labels representing the configurations of the lips, tongue and the combination of glottis and velum.[48] The mapping is presented in Appendix D.[49] The AF-based systems employed feature maps as described in Chapter 5 and were trained to optimize AUC using the algorithm described in Figure 4.5. The system employs three effective articulatory feature streams, each of which is modeled using 3-state labels.

We consider a system that allows up to one state of relative asynchrony ($M = 1$) (Disc-Bottleneck-AF-1) as well as a system that enforces complete synchrony amongst the articulatory feature streams ($M = 0$) (Disc-Bottleneck-AF-0). Our results appear in Table 6.4.

[48]The six Cantonese tones were incorporate as a separate stream that was completely synchronized with the glottis-velum label (G). We gratefully acknowledge the contributions of Eric Fosler-Lussier, Joo-Kyung Kim and Yanzhang He in devising the initial mapping from Cantonese phoneme categories to corresponding IPA phonological features and Karen Livescu for helping develop the corresponding mapping from phonemes to articulatory features.

[49]Note that tones are not explicitly indicated in the mapping presented in Appendix D

System	Average AUC Development Set	Average AUC Evaluation Set
HMM-avgACscore	0.616	0.626
HMM-post	0.821	0.844
Disc-Bottleneck-Phone	0.790	0.792
Disc-Bottleneck-AF-0	0.782	0.792^{*}
Disc-Bottleneck-AF-1	0.788	0.787^{*}
Interp-Disc-Phone/Disc-AF-0 ($\mu = 6.4$)	$0.792^{\dagger}$	0.794^{*}
Interp-Disc-Phone/Disc-AF-1 ($\mu = 19.4$)	0.792	0.789^{*}

Table 6.4: Results of Cantonese STD experiments obtained on the development and evaluation sets, reported in terms of averaged AUC. (*) indicates a statistically significant difference ($p \leq 0.05$) as compared to the HMM-post system using a one-tailed Wilcoxon test of signed-ranks. (†) indicates a statistically significant difference ($p \leq 0.05$) as compared to the Disc-Bottleneck-Phone system using a one-tailed Wilcoxon test of signed-ranks.

As can be see in the table, the articulatory feature-based discriminative systems perform comparably with the phone-based discriminative system on the development set, although performance on the evaluation set is slightly worse. There does not however seem to be any significant difference in performance between the articulatory feature-based systems in terms of whether or not they allow for asynchrony between the articulatory feature streams. This observation is in line with the experiments presented in Chapter 5, where we did not observe significant differences between the AF-based systems in terms of whether or not they allow for articulatory asynchrony.

We also consider interpolation of the scores from the discriminative phone and feature based system,

$$f_{\mathbf{w}}{}^{\text{AF+Ph}}(\overline{\mathbf{x}}, \overline{v}) = f_{\mathbf{w}}{}^{\text{AF}}(\overline{\mathbf{x}}, \overline{v}) + \mu f_{\mathbf{w}}{}^{\text{Ph}}(\overline{\mathbf{x}}, \overline{v}) \tag{6.15}$$

where μ is the interpolation weight. As can be seen, interpolation between the phone and feature based systems result in a small gain relative to the phone and feature-based systems.

On the development set, the difference between Disc-Bottlneck-Phone and Interp-Disc-Phone/Disc-AF-0 is significant ($p = 0.01$) but not the difference between Disc-Bottlenck-Phone and Interp-Disc-Phone/Disc-AF-1 ($p = 0.064$). There is no difference between the discriminative systems in terms of performance on the evaluation set, either with or without interpolation, with all systems performing worse than the baseline HMM-Post system.

6.8 STD Experiments on Switchboard vs. Cantonese

It is interesting to compare the results presented in Chapters 4 and 5 on Switchboard with those presented in this Chapter on Cantonese. There were two main differences in the experimental setups between the English and Cantonese experiments. The first is related to the nature of the the query terms. In our Switchboard experiments, we selected as query terms words that contained at least 5 phonemes in their canonical pronunciation. The words in Cantonese (in general), however, are much shorter (often mono-syllabic) than English words. It is therefore possible that STD for Cantonese is more challenging because there is less phonetic context in the terms. The second difference lies in the cardinality of the phoneme sets ($\sim$50 for English; 130 for Cantonese) and hence the increase in the number of parameters to be estimated. This was particularly an issue with the AF-based systems because of the large number of G configurations owing to our treatment of Cantonese tones.

In comparing the results in the two sets of experiments, we observe that in both cases the discriminative systems performed significantly better than 'acoustic-only' HMM baselines. In the Cantonese dataset, where we use a much stronger baseline, trained on an order of magnitude more data than the discriminative systems, we found that combining the baseline and discriminative systems resulted in large improvements over the baseline. Arguably, the

results presented in these experiments validate the proposed discriminative STD approach presented in this thesis.

The main difference between the results of the experiments in the two setups, however, was related to the performance of the STD systems employing articulatory feature-based pronunciation models. In our Switchboard experiments, we observed gains over the discriminative phone-based system in some settings, particularly when the two systems were interpolated. In the Cantonese experiments, however, we did not see significant improvements over the discriminative phone-based system (except for the interpolated Disc-Phone and Disc-AF-0 system on the development set, where we saw a small statistically significant improvement.) As we have mentioned previously, it is possible that the difference in performance of the two systems may be related to the length of the query terms (with respect to the number of phonemes in their canonical pronunciations). It is relatively straightforward to directly score multi-word query terms within our discriminative STD models, thus providing additional phonetic context to the model. Additionally, such an approach would allow the system to model cross-word asynchrony effects, which might lead to improved performance.

6.9 Relationship of Proposed STD Techniques to ATWV Optimization

Before concluding this chapter, we note that the techniques developed in this thesis were concerned with optimizing the expected area under the receiver operating characteristic. In this section, we provide a brief sketch describing how the techniques proposed in this thesis might apply to directly optimizing the average term weighted value (ATWV) [IAR, 2011; Fiscus et al., 2007] which is the evaluation metric for the BABEL IARPA program.

6.9.1 Average Term Weighted Value: ATWV

Unlike the AUC used in our experiments, the term weighted value (TWV) and average term weighted value (ATWV) are defined in terms of a user-defined threshold, θ_0. For a given term $\overline{v}$ and the threshold, θ, the TWV is defined as [Fiscus et al., 2007],

$$\text{TWV}(\overline{v},\theta) = 1 - \underset{\overline{v}}{\text{average}}\left\{P_{\text{Miss}}(\overline{v},\theta) + \beta P_{\text{FA}}(\overline{v},\theta)\right\} \tag{6.16}$$

$$= \underset{\overline{v}}{\text{average}}\left\{\frac{\text{N}_{\text{correct}}(\overline{v},\theta)}{\text{N}_{\text{true}}(\overline{v})} - \beta\frac{\text{N}_{\text{spurious}}(\overline{v},\theta)}{\text{N}_{\text{NT}}(\overline{v})}\right\} \tag{6.17}$$

where P_{Miss} and P_{FA} are the miss and false-alarm rates, β is a term-dependent scalar that is a measure of the trade-off between the miss rate and the false-alarm rate, $\text{N}_{\text{correct}}(\overline{v},\theta)$ is the number of correct detections of $\overline{v}$ with a score greater than or equal to θ, $\text{N}_{\text{spurious}}(\overline{v},\theta)$ is the number of incorrect detections with a score greater than or equal to θ, $\text{N}_{\text{true}}(\overline{v})$ is the number of occurrences of the term in the corpus, and $\text{N}_{\text{NT}}(\overline{v}) = \text{T}_{\text{speech}} - \text{N}_{\text{true}}(\overline{v})$, with T_{speech} being the size of the corpus in seconds.

The ATWV is then defined as the average TWV over all of the query terms,

$$\text{ATWV} = \underset{\overline{v}}{\text{average}}\,\text{TWV}(\overline{v},\theta_0(\overline{v})) \tag{6.18}$$

where $\theta_0(\overline{v})$ is the (term-dependent) threshold chosen by the user.

6.9.2 Training for the Cantonese Babel Data

We may utilize information in the index, $\mathcal{I}$, generated from the baseline system to produce a set of candidate word locations, $\mathcal{X} = \{\overline{\mathbf{x}}_i, \overline{v}_i, P(\overline{v}_i|\overline{\mathbf{x}}_i)\}_{i=1}^{N}$, where $\overline{\mathbf{x}}_i$ is the set of frames corresponding to the hypothesized word location, $\overline{v}_i$ is the hypothesized word, and $P(\overline{v}_i|\overline{\mathbf{x}}_i)$ is the posterior probability computed by the baseline system. Every entry in $\mathcal{X}$ will correspond to either a *hit* or a *false alarm*. In order to indicate whether a hypothesized

word is a hit or false alarm, we define the function $\delta^+(\overline{\mathbf{x}}, \overline{v}) = 1$ if and only if $\overline{\mathbf{x}}$ is a hit for the term $\overline{v}$ and 0 otherwise and analogously define the function $\delta^-(\overline{\mathbf{x}}, \overline{v}) = 1$ if and only if $\overline{\mathbf{x}}$ is a false alarm for the term $\overline{v}$ and 0 otherwise. Finally, let $X^+(\overline{v})$ denote the number of positive examples of a term $\overline{v}$ in the training data. Using our linear STD model, the average term weighted value, ATWV(θ), for the training data is given by,[50]

$$\text{ATWV}(\theta) = \frac{1}{|V|} \sum_{i=1}^{N} \left\{ \frac{1}{X^+(\overline{v}_i)} \delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta] \delta^+(\overline{\mathbf{x}}_i, \overline{v}_i) - \frac{\beta}{\text{T}_{\text{speech}} - X^+(\overline{v}_i)} \delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta] \delta^-(\overline{\mathbf{x}}_i, \overline{v}_i) \right\} \tag{6.19}$$

Setting $\beta'(\overline{v}) = \frac{\beta}{\text{T}_{\text{speech}} - X^+(\overline{v})}$, the optimal weight vector that maximizes ATWV(θ) in Equation 6.19 is given by,

$$\mathbf{w}^*(\theta) = \underset{\mathbf{w}}{\text{argmax}} \sum_{i=1}^{N} \left\{ \frac{1}{X^+(\overline{v}_i)} \delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta] \delta^+(\overline{\mathbf{x}}_i, \overline{v}_i) - \beta'(\overline{v}_i) \delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta] \delta^-(\overline{\mathbf{x}}_i, \overline{v}_i) \right\} \tag{6.20}$$

$$= \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{N} \left\{ \frac{1}{X^+(\overline{v}_i)} (1 - \delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta]) \delta^+(\overline{\mathbf{x}}_i, \overline{v}_i) + \beta'(\overline{v}_i) (\delta[f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i) > \theta]) \delta^-(\overline{\mathbf{x}}_i, \overline{v}_i) \right\} \tag{6.21}$$

Instead of optimizing Equation 6.21 directly, which is a non-smooth problem, we consider the following function which is a smooth upper bound to the function in Equation 6.21,

$$\mathbf{w}^*(\theta) = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} \left\{ \frac{1}{X^+(\overline{v}_i)} [1 + \theta - f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i)]_+ \delta^+(\overline{\mathbf{x}}_i, \overline{v}_i) + \beta'(\overline{v}_i) [1 - \theta + f_{\mathbf{w}}(\overline{\mathbf{x}}_i, \overline{v}_i)]_+ \delta^-(\overline{\mathbf{x}}_i, \overline{v}_i) \right\} \tag{6.22}$$

Equation 6.22, can now be optimized using the MM algorithm [Hunter and Lange, 2004], analogously to the algorithm presented in Figure 4.5. We are currently in the process of evaluating whether the techniques briefly sketched in this section are effective for optimizing expected ATWV.

6.10 Relationship Between AUC and TWV

Apart from training a system directly to optimize ATWV as described in Section 6.9.2, it may also be possible to exploit the following relationship that exists between the AUC

[50] Where we assume a fixed threshold θ for all of the query terms.

and the TWV for a particular term $\overline{v}$. Let us denote by AUC($\overline{v}$) the AUC of the classifier for a particular term. We denote by TPR($\overline{v}, \theta$) and FPR($\overline{v}, \theta$) the true positive rate and the false positive rate of the classifier at a threshold $\theta \in \mathbb{R}$ for the terms $\overline{v}$, respectively, and by $\text{FPR}^{-1}(\overline{v}, x)$, the lowest threshold which results in a false positive rate of $x \in [0, 1]$ for the classifier when detecting term $\overline{v}$,

$$\text{FPR}^{-1}(\overline{v}, x) = \inf\{\theta \in \mathbb{R} : \text{FPR}(\overline{v}, \theta) \leq x\} \tag{6.23}$$

With these definitions we can write,

$$\text{AUC}(\overline{v}) = \int_0^1 \text{TPR}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x)) \quad dx \tag{6.24}$$

Finally, we denote the term weighted value for a particular term $\overline{v}$ at a threshold θ as TWV($\overline{v}, \theta$). With these definitions, consider the term weighted value of the classifier for a particular term $\overline{v}$ averaged over all possible false positive rates, which we denote by FP-TWV($\overline{v}$),

$$\text{FP-TWV}(\overline{v}) = \int_0^1 \text{TWV}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x)) \quad dx \tag{6.25}$$

$$= \int_0^1 \left\{ \frac{\text{N}_{\text{correct}}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x))}{\text{N}_{\text{true}}(\overline{v})} - \beta \frac{\text{N}_{\text{spurious}}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x))}{\text{T}_{\text{speech}} - \text{N}_{\text{true}}(\overline{v})} \right\} \quad dx \tag{6.26}$$

where Equation 6.26 follows from Equation 6.17.

But $\frac{\text{N}_{\text{correct}}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x))}{\text{N}_{\text{true}}(\overline{v})} = \text{TPR}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x))$; $\text{N}_{\text{spurious}}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x)) = |\mathcal{X}^-(\overline{v})|\, x$, where $|\mathcal{X}^-(\overline{v})|$ is the number of candidate entries in the speech index for the term $\overline{v}$ which are not hits for the term. Thus,

$$\text{FP-TWV}(\overline{v}) = \int_0^1 \left\{ \text{TPR}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x)) - \frac{\beta\, |\mathcal{X}^-(\overline{v})|}{\text{T}_{\text{speech}} - \text{N}_{\text{true}}(\overline{v})} x \right\} \quad dx \tag{6.27}$$

$$= \int_0^1 \text{TPR}(\overline{v}, \text{FPR}^{-1}(\overline{v}, x)) \quad dx - \int_0^1 \left\{ \frac{\beta\, |\mathcal{X}^-(\overline{v})|}{\text{T}_{\text{speech}} - \text{N}_{\text{true}}(\overline{v})} x \right\} \quad dx \tag{6.28}$$

$$= \text{AUC}(\overline{v}) - \frac{\beta\, |\mathcal{X}^-(\overline{v})|}{2(\text{T}_{\text{speech}} - \text{N}_{\text{true}}(\overline{v}))} \tag{6.29}$$

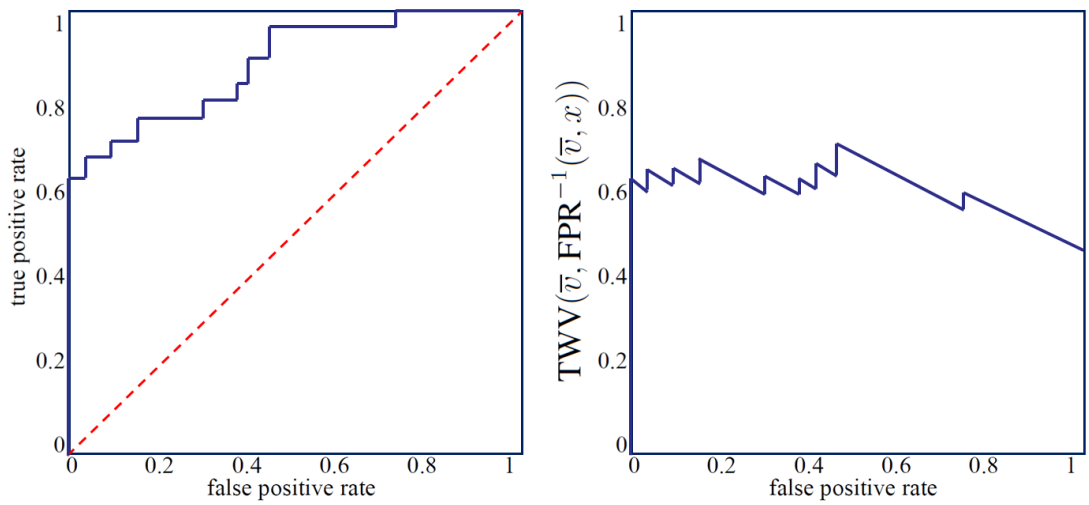

Figure 6.3: The figure on the left illustrates the AUC for a particular term. The figure on the right illustrates the TWV(FP$^{-1}(\overline{v}, x)$) as a function of the false positive rate (x). The model parameters that maximize AUC also maximize the area under the curve on the right.

Finally, note that the second term in Equation 6.29, $\frac{\beta|\mathcal{X}^-(\overline{v})|}{2(\mathrm{T_{speech}} - \mathrm{N_{true}}(\overline{v}))}$, is independent of the parameters of the classifier. Therefore, the set of parameters, $\mathbf{w}^*$, that optimize AUC for a particular term are the same as the parameters that optimize TWV$(\overline{v}, \mathrm{FPR}^{-1}(\overline{v}, x))$ averaged over the range of false positive rates. In other words, across the range of thresholds, each of which corresponds to a particular false positive rate, a system trained to optimize AUC will, on average (across thresholds), yield high TWV performance for a particular term. Similarly, across the range of possible thresholds for each of the terms, a system that is trained to optimize average AUC across the terms will yield a system that yields high ATWV performance averaged across the terms. The relationship between AUC and TWV for a particular term is illustrated in Figure 6.3.

6.10.1 Pilot Experiments: Evaluating System Trained to Optimize AUC in Terms of ATWV

The relationship between ATWV and AUC described in Section 6.10 provides us with a framework for developing systems that yield high performance in terms of ATWV. Essentially, we first train a system to optimize AUC, and then tackle the problem of finding a term-dependent threshold, $\theta_0(\overline{v})$, that is likely to yield high ATWV performance.

In this section, we describe a set of pilot experiments which use a simple technique for estimating the threshold $\theta_0(\overline{v})$ using the baseline system. We construct a baseline Cantonese STD system following [Miller et al., 2007], which declares a candidate entry in the speech index $\left\langle \overline{\mathbf{x}}_i, \overline{v}, s_i^1, e_i^M, P(\overline{v}|\overline{\mathbf{x}}_i) \right\rangle \in \text{Cand}(\overline{v})$ as a putative detection of the term $\overline{v}$ if,

$$P(\overline{v}|\overline{\mathbf{x}}_i) \geq \frac{\mathrm{N}_{\text{true}}}{\frac{\mathrm{T}_{\text{speech}}}{\beta} + \frac{\beta-1}{\beta}\mathrm{N}_{\text{true}}} \tag{6.30}$$

Let $\mathcal{X}^+(\overline{v})$ be the set of candidate terms that are declared to be hits according to Equation 6.30. Given a discriminative system trained to optimize AUC, we evaluate the system in terms of its ATWV performance by choosing as the threshold $\theta_0(\overline{v})$ so that the discriminative system returns the same number of candidates $|\mathcal{X}^+(\overline{v})|$ as the baseline. In the following set of experiments, we only evaluate the two discriminative systems Interp-HMMpost/Bottleneck ($\mu = 15.5$) and Interp-HMMpost/Bottleneck-impNegSel ($\mu = 19.4$) which appeared in Section 6.6 and which were the best performing systems in terms of AUC. Our results appear in Table 6.5.

As can be seen in Table 6.5, both interpolated discriminative systems perform comparably with the baseline system in terms of ATWV on the development set. On the evaluation set, however, the baseline system performs significantly better than the interpolated system

System	ATWV Dev. Set	ATWV Eval. Set
Baseline (HMM-post) [Miller et al., 2007]	0.414	0.488
Interp-HMMpost/Bottleneck ($\mu = 15.5$)	0.399	0.453*
Interp-HMMpost/Bottleneck-impNegSel ($\mu = 29.1$)	0.426	0.474

Table 6.5: Results of baseline system (HMM-post, described in Section 6.3 with term dependent thresholding [Miller et al., 2007]) and the interpolated discriminative systems trained to optimize AUC, evaluated in terms of their ATWV performance. The interpolated discriminative systems' thresholds are set so that the systems return as putative hits the same number of entries (per query term) as the baseline. (*) denotes a significant difference ($p \leq 0.05$) relative to the Baseline system with term dependent thresholding [Miller et al., 2007] computed using a one-tailed Wilcoxon test of signed-ranks.

Interp-HMMpost/Bottleneck (i.e. the system *without improved selection of negative examples*) ($p = 0.02$); the difference between the baseline and the Interp-HMMpost/Bottleneck-impNegSel system (i.e. the system *with improved selection of negative examples*) is not significant ($p = 0.12$) using a one-tailed Wilcoxon signed-ranks test. Another interesting observation that can be made based on the results in Table 6.5 is that the system with higher AUC performance (Interp-HMMpost/Bottleneck-impNegSel) also has higher ATWV performance; this observation is consistent with the relationship between AUC and ATWV that we outlined in Section 6.10.

In summary, the results in Table 6.5 suggest that although the interpolated discriminative systems perform 3.4–4.8% better than the baseline in terms of AUC performance, they do not significantly outperform the baseline in terms of ATWV performance when we use the simple threshold determination scheme described in this section. One possible explanation of this fact is due to the nature of the difference between ATWV and AUC: the AUC treats all regions of the ROC curve equally; improvements in the true positive rate at all levels of the false positive rate contribute equally to the AUC. The ATWV, on the other hand is strongly biased towards the part of the ROC curve corresponding to low false

positive rates (i.e., the region to the left of the ROC curve appearing in Figure 6.3).[51] Thus, if the improvements in the AUC are a result of improving true positive rates in the part of the ROC curve corresponding to high false positive rates (i.e., the region to the right of the ROC curve appearing in Figure 6.3) the AUC improvement will not result in an ATWV improvement if we use the simple thresholding scheme described in this section. Note, however, that although in Section 6.10 we showed that the parameters of the classifier that maximize AUC are the same as the parameters that maximize ATWV averaged across the range of false positive rates (integrated from 0 to 1), the same relationship holds over any range of false positive rates. Thus, it may be beneficial to adapt our training algorithms for the discriminative systems in order to optimize *partial AUC* in the range of low false positive rates ($[0, f]$, where, $f \ll 1$) [Rudin, 2009; Agarwal, 2011; Rakotomamonjy, 2012], in order to try and ensure improvements in ATWV by focusing on the part of the ROC curve corresponding to the operating point chosen according to the simple thresholding scheme described in this section. We analyze some of these issues further in Section 6.10.2.

6.10.2 Further Analysis of Systems Trained to Optimize AUC Evaluated in Terms of their ATWV Performance

In order to further examine some of the issues that were raised in the previous section, we consider an alternative thresholding scheme for determining the threshold $\theta_0(\overline{v})$ using the baseline system [Miller et al., 2007]: If $\mathcal{X}^+(\overline{v})$ represents the set of candidate terms that are declared to be putative hits according to Equation 6.30, we return the top-scoring $|\mathcal{X}^+(\overline{v})| + \tau$ entries, where $\tau \in \mathbb{Z}$ is an integer, for the query term $\overline{v}$.[52] Note that this new

[51] In fact, generally speaking, the thresholding scheme in Equation 6.30 [Miller et al., 2007] is highly conservative in declaring an entry in the speech index to be a putative hit for the query term. This is especially true for those terms which have a large number of candidates in the speech index.

[52] If $|\mathcal{X}^+(\overline{v})| + \tau < 0$, we return 0 entries. Similarly, if there are fewer candidates than the number required to be returned (i.e., $|\mathcal{X}^+(\overline{v})| + \tau > |\text{Cand}(\overline{v})|$) then we return all of the candidates ($|\text{Cand}(\overline{v})|$).

thresholding scheme can be applied to the baseline system as well as to the interpolated discriminative systems. In Figure 6.4 we plot the performance obstained by returning the top-scoring $|\mathcal{X}^+(\overline{v})|+\tau$ entries for the baseline (HMM-post), and the two interpolated discriminative systems Interp-HMMpost/Bottleneck and Interp-HMMpost/Bottleneck-impNegSel, as a function of τ.

As can be seen in Figure 6.4, the performance of both interpolated systems is comparable to the baseline for $\tau \leq 1$. However, for larger values of $\tau \geq 2$, the interpolated discriminative systems outperform the baseline in terms of ATWV. Recall that under the ATWV metric, each false alarm for a term $\overline{v}$ results in a penalty given by $\frac{\beta}{\mathrm{T_{speech}}-\mathrm{N_{true}}(\overline{v})}$. Since the size of the corpus in seconds is much larger than the number of occurrences of the query term in the corpus (i.e., $\mathrm{T_{speech}} \gg \mathrm{N_{true}}(\overline{v})$), the penalty for each false alarm is essentially term-independent and is given by $\frac{\beta}{\mathrm{T_{speech}}}$. Since for any fixed τ, we return the same number of total entries for either the baseline system or for the interpolated discriminative systems under the thresholding scheme described in this section, the fact that the interpolated discriminative systems score higher than the baseline system in terms of ATWV is indicative of the fact that on average (across query terms) these systems tend to score positive examples higher than negative examples in the set of candidates for any query term. Furthermore, since the difference between the systems is larger for higher values of τ, we conclude that the difference in AUCs between the interpolated discriminative systems and the baseline is greater in the range of higher false positive rates than in the range of lower false positive rates. Thus, these results are consistent with our hypothesis in the previous section: further ATWV improvements might be obtained by focusing on improving partial AUC in the range of low false positive rates [Rudin, 2009; Agarwal, 2011; Rakotomamonjy, 2012]. For completeness, in Table 6.6 we list system performance obtained

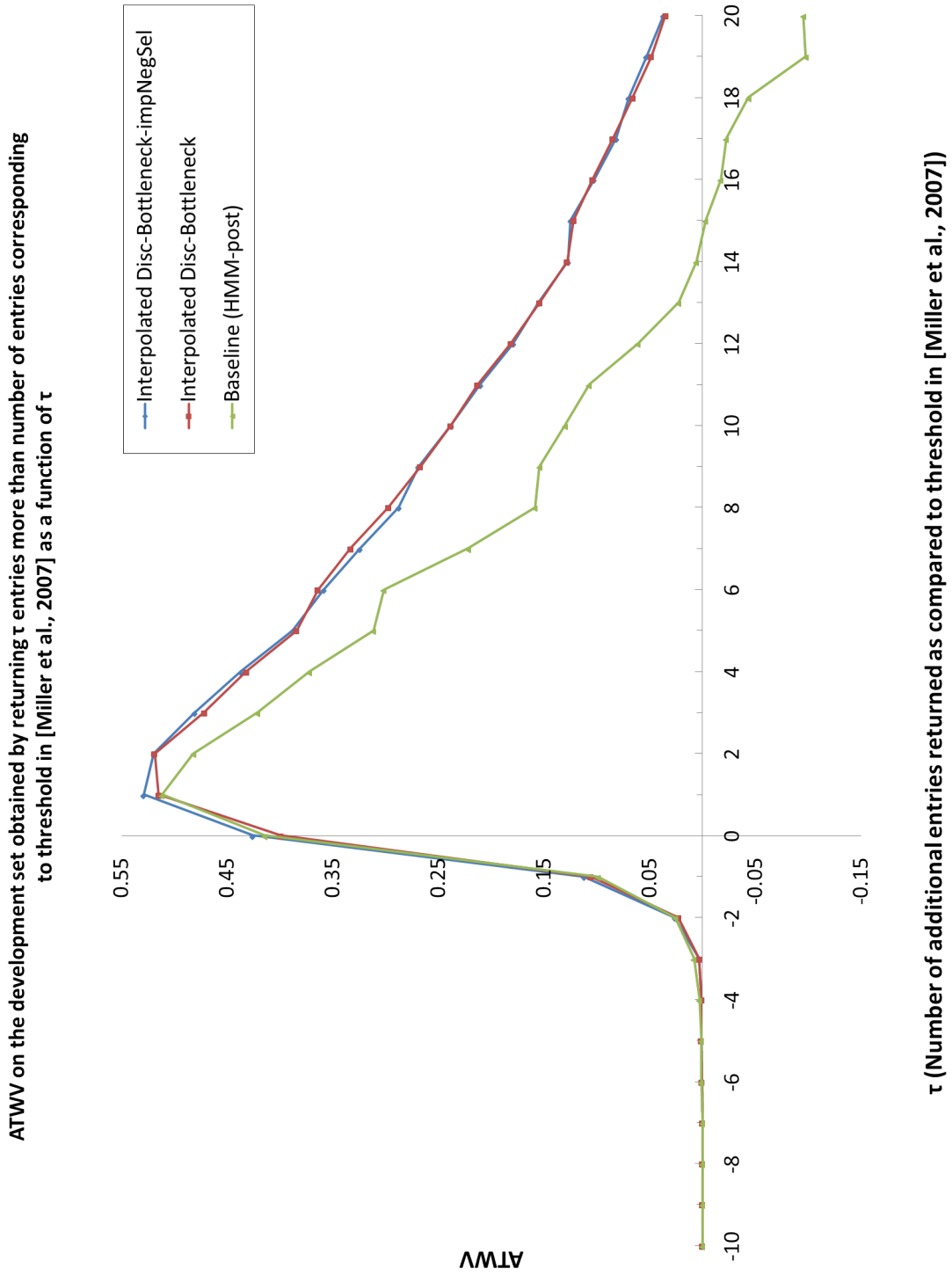

Figure 6.4: Performance of the baseline as well as the interpolated discriminative systems in terms of ATWV obtained by returning the top-scoring $|\mathcal{X}^+(\overline{v})| + \tau$ candidates for query term $\overline{v}$, where $\mathcal{X}^+(\overline{v})$ represents the set of candidate terms that are declared to be putative hits according to Equation 6.30, as a function of τ in the range $[-10, 20]$.

System	ATWV Dev. Set	ATWV Eval. Set
Baseline (HMM-post) [Miller et al., 2007]	0.414	0.488
Baseline (HMM-post) ($\tau = 1$)	0.512	0.535
Interp-HMMpost/Bottleneck ($\mu = 15.5$) ($\tau = 2$)	0.519	$0.428^{*,\dagger}$
Interp-HMMpost/Bottleneck-impNegSel ($\mu = 29.1$) ($\tau = 1$)	$0.529^{*,\dagger}$	0.546

Table 6.6: Results of baseline system (HMM-post, described in Section 6.3 with term dependent thresholding [Miller et al., 2007]; this corresponds to setting $\tau = 0$) and systems obtained by returning the top-scoring $|\mathcal{X}^+(\overline{v})| + \tau$ entries, where $\mathcal{X}^+(\overline{v})$ is the set of entries returned by the baseline according to Equation 6.30 [Miller et al., 2007] for the best value of τ tuned on the development set. (*) and (†) denote significant differences ($p \leq 0.05$) relative to the baseline system with term dependent thresholding [Miller et al., 2007] (row 1 in Table 6.6) and the baseline system that returns $|\mathcal{X}^+(\overline{v})| + 1$ entries for each term (row 2 in Table 6.6), respectively, computed using a one-tailed Wilcoxon test of signed-ranks.

on the development and evaluation sets for the best value of τ (obtained by tuning on the development set) for the baseline as well as the discriminative interpolated systems.

6.11 Summary

In this chapter, we presented experiments for STD that explicitly leverage existing LVCSR-based STD systems for discriminative STD. In experimental results, we found that our proposed acoustic-only STD systems did not perform as well as the baseline HMM systems which use word-level context. However, interpolating discriminative system scores with the baseline posterior score resulted in large gains (in terms of averaged AUC) over the baseline. We also experimented with the use of discriminative AF-based pronunciation models, where we found that the AF-based system performance was similar to the phone-based discriminative systems.

CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we have presented a series of experiments that were aimed at determining the effectiveness of using articulatory feature-based pronunciation models in order to address the increased variability in conversational speech. In the first part of the thesis, we developed discriminative conditional random field (CRF) models for the task of articulatory feature forced transcription. One of the contributions of this thesis was to demonstrate how deterministic task-specific constraints allowed for efficient exact inference in the model. In experimental evaluations, we found that the proposed models outperformed previously proposed dynamic Bayesian network (DBN) models for the task [Livescu and Glass, 2004a,b]. In the second part of the thesis, we extended previous work on discriminative spoken term detection [Keshet et al., 2009], which allowed us to incorporate the proposed articulatory feature-based pronunciation models within a spoken term detection system. In experimental evaluations in low-resource settings, we observed that the proposed articulatory feature-based STD systems outperformed baseline hidden Markov model-based (HMM-based) STD systems as well as discriminative phone-based systems in various settings. Finally, in the last part of the thesis, we demonstrated how large vocabulary continuous speech recognizer-based (LVCSR-based) STD systems could be leveraged in order to improve performance as well as running time of our proposed STD systems.

Our experiments and analyses revealed a number of interesting observations. For example, in our experiments on articulatory feature forced transcription, we found that the performance of the *generative* DBNs did not improve if we allowed the articulatory feature

streams to transition asynchronously. This finding was contrary to the findings of previous lexical access experiments [Livescu and Glass, 2004a,b], where allowing for articulatory asynchrony resulted in large gains. Similarly, when the articulatory feature-based pronunciation models were incorporated within an STD system, we did not see a significant difference in system performance between an articulatory feature-based system that allowed for up to one state of relative asynchrony and an articulatory feature-based system that enforced complete synchrony amongst the articulatory feature streams. However, although the two systems performed comparably, our analysis of the asynchronous AF-based system indicated that the system appeared to be hypothesizing additional asynchrony for those examples that appeared to contain additional pronunciation variation. This results seems to suggest that the AF-based systems do indeed model some of the pronunciation variability in the speech.

In our experiments on Cantonese STD, presented in Chapter 6, we compared performance obtained from our acoustic-only STD systems to a very strong discriminative LVCSR-based system trained on an order of magnitude more data. In these experiments, we found that combining our discriminative system with the baseline posterior score resulted in large improvements for the STD systems. The results of our experiments on discriminative STD, conducted on both Switchboard as well as the Cantonese data serve to validate the proposed approach. However, in our pilot experiments when we applied AF-based pronunciation models for Cantonese STD, we did not seem significant improvements (or degradation) in performance relative to the discriminative phone-based systems. Finally, we discussed techniques by which the proposed techniques for optimizing AUC can be adapted for optimizing ATWV [Fiscus et al., 2007] and we also discussed how

the relationship between AUC and ATWV can be exploited to evaluate systems trained to optimize AUC in terms of their ATWV performance.

7.1 Future Work

We end this thesis with a description of some of the future research themes suggested by the work presented in this thesis:

- **Incorporation of AF-based models with AF substitution:** The theory of articulatory phonology [Browman and Goldstein, 1992] suggests that the pronunciation variation observed in conversational speech can be accounted for by two processes: (a.) gestural overlaps and (b.) reductions in gestural magnitudes. The models described in this thesis incorporate a mechanism for modeling gestural overlaps, but do not account for reduction in gestural magnitudes through articulatory feature substitution [Livescu, 2005]. Our main motivation for this was to avoid the additional complexity of the AF substitution model. Enriching our models by incorporating a model of AF substitution, without tremendously increasing model complexity, is an interesting research direction.

- **Richer model of articulatory feature asynchrony:** In Section 5.4, we analyzed the percentage of asynchronous frames hypothesized by AF-based systems that allowed for up to either one ($M = 1$) or two ($M = 2$) states of asynchrony, and found that the system which allows for additional asynchrony hypothesizes lower amounts of asynchrony in both positive as well as negative examples of the query term. We hypothesized that the system which allows for up to two frames of asynchrony allows for greater flexibility in finding high scoring segmentations for the negative examples, which might cause the model to learn to hypothesize lower amounts of asynchrony

in order to minimize confusability. This would suggest that the proposed AF-based pronunciation models might benefit from a richer model of articulatory asynchrony, which might help constrain the set of articulatory segmentations and allow the model to learn to only hypothesize articulatory asynchrony in linguistically plausible contexts.

- **Discriminative STD optimizing other measures of task performance:** The models presented in this thesis extended work by [Keshet et al., 2009] by training a discriminative STD system to optimize area under the ROC curve (i.e. AUC). There are a number of other metrics commonly used for evaluation of STD performance such as figure-of-merit (FoM) [Wallace et al., 2011] and average term weighted value (ATWV) [Fiscus et al., 2007]. The models presented in this thesis could be adapted to directly optimize these alternative metrics (we provided a brief sketch of this idea in Sections 6.9 and 6.10) and we leave this as a promising future research direction.

- **Incorporation of discriminative AF-based pronunciation model within an end-to-end speech recognizer:** The articulatory feature-based models of pronunciation described in this thesis are *discriminative* as opposed to the *generative* models investigated in prior research [Livescu, 2005; Livescu et al., 2007a]. When these generative models were incorporated within an end-to-end speech recognizer by Livescu et al., they did not result in large improvements over a baseline monophone system on a medium vocabulary SVitchboard task [King et al., 2005]. Given the dominance of discriminatively trained HMM systems in modern ASR, as well as the successes of the discriminative STD systems presented in this thesis, it would be interesting to apply these models as part of a discriminative end-to-end speech recognizer. Of

course, moving from the current 'verification-based' STD approach presented in this thesis to a general word recognition approach is challenging and would likely require significant research effort. One possibility to deal with the increased complexity is to implement the AF-based models within the segmental conditional random field (SCARF) paradigm of Zweig and Nguyen [2010].

7.2 Contributions of the Thesis

We end this thesis by re-stating the main contributions of this thesis: the development of discriminative articulatory feature-based pronunciation models, and the application of these models to the task of spoken term detection for conversational speech. The contributions of this thesis are:

- **Discriminative Articulatory Feature-based Pronunciation Modeling:** We developed discriminative articulatory feature-based pronunciation models using conditional random fields and applied these models to the task of extracting articulatory features from speech utterances given their word transcriptions. In particular, we demonstrated how the deterministic task-specific constraints that exist in our problem can be exploited to perform exact inference efficiently in our models. In experimental evaluations, we found that the proposed models outperform previously proposed generative dynamic Bayesian network models for the task.

- **Discriminative Spoken Term Detection in Low-Resource Settings:** We applied our discriminative articulatory feature-based pronunciation models within a discriminative spoken term detection system extending previous work [Keshet et al., 2009] and evaluated these models in the setting of limited training data. In experimental

evaluations, we found that our proposed discriminative systems outperformed baseline GMM-HMM systems by large margins across a range of training set sizes.

- **Discriminative Spoken Term Detection Leveraging Existing LVCSR-based Systems:** We described how our proposed approach for training discriminative systems can be adapted in order to both speed up the training process of our discriminative STD systems as well as to improve system performance by leveraging existing LVCSR-based STD systems. In experimental evaluations on a subset of the IARPA Babel Cantonese data [IAR, 2011], we found that combining discriminative systems with the baseline system resulted in large performance improvements over the baseline in terms of AUC.

The work presented in this thesis has raised a number of interesting research questions and suggested many promising research directions for both articulatory feature-based pronunciation modeling and spoken term detection. We hope that future work in these areas will help improve system performance further, thus bringing computational systems closer to human performance in challenging problem domains such as the recognition of conversational speech.

APPENDIX A: TRAINING ALGORITHM FOR DISCRIMINATIVE SPOKEN TERM DETECTION VIEWED AS AN INSTANCE OF THE CONVEX-CONCAVE PROCEDURE

In this chapter, we show how the algorithm for optimizing Equation 4.8, which appears in Figure 4.5, can be viewed as an instance of the Convex-Concave Procedure [Yuille and Rangarajan, 2002].

A.1 A Brief Overview of the Convex-Concave Procedure

The Convex-Concave Procedure (CCCP) [Yuille and Rangarajan, 2002] is a technique for optimizing a function $f(\boldsymbol{\theta})$ iteratively, given a current estimate $\boldsymbol{\theta}_m \in \mathbb{R}^d$. CCCP can be applied to minimize functions which can be written as a sum of two functions $g^{\text{vex}}(\boldsymbol{\theta})$ and $g^{\text{cave}}(\boldsymbol{\theta})$ which are respectively convex and concave with respect to the parameters $\boldsymbol{\theta}$ over the domain $\mathbb{R}^d$.[53] In other words, CCCP is an optimization technique to solve problems of the form,

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} g^{\text{vex}}(\boldsymbol{\theta}) + g^{\text{cave}}(\boldsymbol{\theta}) \tag{A.1}$$

where $g^{\text{vex}}(\boldsymbol{\theta})$ is convex and $g^{\text{cave}}(\boldsymbol{\theta})$ is concave.

CCCP exploits the following property of a concave function which holds for any fixed $\boldsymbol{\theta}_0 \in \mathbb{R}^d$: the plane that is tangent to the concave function at $\boldsymbol{\theta}_0$ lies above the surface of the function for all $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$g^{\text{cave}}(\boldsymbol{\theta}) \leq g^{\text{cave}}(\boldsymbol{\theta}_0) + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T \nabla g^{\text{cave}}(\boldsymbol{\theta}_0) \tag{A.2}$$

[53] In fact, any function f(x) with bounded Hessian can be written as a sum of a convex and a concave function [Yuille and Rangarajan, 2002].

Exploiting Equation A.2, we can replace the concave function in Equation A.1 with the tangent plane at the current estimate of the minimum $\boldsymbol{\theta}_m$ (i.e. substituting $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_m$ in Equation A.2) to obtain a new function that is a convex upper bound of the original function for all $\boldsymbol{\theta} \in \mathbb{R}^d$. Denoting this function by $f^{\text{ub}}(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$, we have,

$$f^{\text{ub}}(\boldsymbol{\theta}; \boldsymbol{\theta}_m) = g^{\text{vex}}(\boldsymbol{\theta}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_m)^T \nabla g^{\text{cave}}(\boldsymbol{\theta}_m) \geq f(\boldsymbol{\theta}) \tag{A.3}$$

The convex-concave procedure proceeds by minimizing the convex upper bound, $f^{\text{ub}}(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$, instead of the original function $f(\boldsymbol{\theta})$. Denoting the the minimizer of $f^{\text{ub}}(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ as $\boldsymbol{\theta}_{m+1}$, we note that the gradient of the upper bound function at the minimum value must be equal to zero. Setting $\nabla f^{\text{ub}}(\boldsymbol{\theta}_{m+1}; \boldsymbol{\theta}_m) = 0$, we derive the following update equation which must be satisfied by $\boldsymbol{\theta}_{m+1}$:

$$\nabla g^{\text{vex}}(\boldsymbol{\theta}_{m+1}) = -\nabla g^{\text{cave}}(\boldsymbol{\theta}_m) \tag{A.4}$$

Thus, Equation A.4 provides us with an iterative procedure for estimating a new parameter estimate $\boldsymbol{\theta}_{m+1}$ given the current estimate $\boldsymbol{\theta}_m$. Finally, we show that this new estimate $\boldsymbol{\theta}_{m+1}$ in Equation A.4 has a lower value of the original objective function $f(\boldsymbol{\theta})$. Since $\boldsymbol{\theta}_{m+1}$ minimizes $f^{\text{ub}}(\boldsymbol{\theta}; \boldsymbol{\theta}_m)$ we can write,

$$f^{\text{ub}}(\boldsymbol{\theta}_m; \boldsymbol{\theta}_m) \geq f^{\text{ub}}(\boldsymbol{\theta}_{m+1}; \boldsymbol{\theta}_m) \tag{A.5}$$

$$\therefore g^{\text{vex}}(\boldsymbol{\theta}_m) \geq g^{\text{vex}}(\boldsymbol{\theta}_{m+1}) + (\boldsymbol{\theta}_{m+1} - \boldsymbol{\theta}_m)^T \nabla g^{\text{cave}}(\boldsymbol{\theta}_m) \quad \text{(from Equation A.3)} \tag{A.6}$$

Setting $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_m$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_{m+1}$ in Equation A.2, we have from Equation A.6,

$$g^{\text{vex}}(\boldsymbol{\theta}_m) \geq g^{\text{vex}}(\boldsymbol{\theta}_{m+1}) + g^{\text{cave}}(\boldsymbol{\theta}_{m+1}) - g^{\text{cave}}(\boldsymbol{\theta}_m) \tag{A.7}$$

$$\therefore g^{\text{vex}}(\boldsymbol{\theta}_m) + g^{\text{cave}}(\boldsymbol{\theta}_m) \geq g^{\text{vex}}(\boldsymbol{\theta}_{m+1}) + g^{\text{cave}}(\boldsymbol{\theta}_{m+1}) \tag{A.8}$$

$$\iff f(\boldsymbol{\theta}_m) \geq f(\boldsymbol{\theta}_{m+1}) \tag{A.9}$$

Thus, $\boldsymbol{\theta}_{m+1}$ has a lower value of the objective function $f(\boldsymbol{\theta})$ than the current estimate $\boldsymbol{\theta}_m$. In summary, CCCP proceeds by iteratively solving Equation A.4 until the procedure converges to a local minimum of the objective function [Yuille and Rangarajan, 2002].

A.2 Viewing Algorithm in Figure 4.5 as an Instance of CCCP

Finally, we end this chapter by demonstrating how our algorithm in Figure 4.5 can be viewed as an instance of CCCP. Recall that the original non-convex problem that we would like to solve appeared in Equation 4.8:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left[1 - f_{\mathbf{w}}(\overline{\mathbf{x}}_i^+, \overline{v}_i) + f_{\mathbf{w}}(\overline{\mathbf{x}}_i^-, \overline{v}_i)\right]_+ \tag{A.10}$$

$$= \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left[1 - \max_{\overline{\mathbf{s}}\sim(s_i^+, e_i^+)} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}) + \max_{\overline{\mathbf{s}}} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right]_+ \tag{A.11}$$

where $[0, x]_+ = \max\{0, x\}$. We can re-write Equation A.11 as follows,

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\max\left\{0, 1 - \max_{\overline{\mathbf{s}}\sim(s_i^+, e_i^+)} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}) + \max_{\overline{\mathbf{s}}} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\} \tag{A.12}$$

$$= \operatorname*{argmin}_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\max\left\{\max_{\overline{\mathbf{s}}\sim(s_i^+, e_i^+)} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}), 1 + \max_{\overline{\mathbf{s}}} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\}\right)$$
$$- \left(\frac{1}{N}\sum_{i=1}^{N}\max_{\overline{\mathbf{s}}\sim(s_i^+, e_i^+)} \mathbf{w}\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}})\right) \tag{A.13}$$

Notice that in Equation A.13, the first term is convex, while the second term is concave. We denote by $\mathbf{w}_t$ the estimate of the optimal weight vector at the end of the t-th epoch and by $\overline{\mathbf{s}}_i^+(\mathbf{w}_t)$ the segmentation of the i-th positive example that scores highest when the weight vector corresponds to $\mathbf{w}_t$,

$$\overline{\mathbf{s}}_i^+(\mathbf{w}_t) = \operatorname*{argmax}_{\overline{\mathbf{s}}\sim(s_i^+, e_i^+)} \mathbf{w}_t\cdot\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}) \tag{A.14}$$

In order to optimize the problem in Equation A.13, we linearize the concave part of the optimization problem at the current estimate of the weight vector using the sub-gradient as in Equation A.3 to obtain an improved estimate, $\mathbf{w}_{t+1}$, of the optimal weight vector:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} \max \left\{ \max_{\bar{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}), 1 + \max_{\bar{\mathbf{s}}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \bar{\mathbf{s}}) \right\} \right) - \left(\frac{1}{N}\sum_{i=1}^{N} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t)) \right) \tag{A.15}$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} \max \left\{ \max_{\bar{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}) - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t)), 1 - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t)) + \max_{\bar{\mathbf{s}}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \bar{\mathbf{s}}) \right\} \tag{A.16}$$

Finally, we make the approximation that the difference $\max_{\bar{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}) - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t))$ is approximately 0,[54] so that we can once again re-write the optimization problem in terms of a hinge loss as follows:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} \max \left\{ 0, 1 - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t)) + \max_{\bar{\mathbf{s}}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \bar{\mathbf{s}}) \right\} \tag{A.17}$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} [1 - \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \bar{\mathbf{s}}_i^+(\mathbf{w}_t)) + \max_{\bar{\mathbf{s}}} \mathbf{w} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \bar{\mathbf{s}})]_+ \tag{A.18}$$

Thus, we notice that the final optimization problem that we have derived in Equation A.18 corresponds exactly to the majorizer that we derived in Equation 4.18. Thus, the algorithm that appears in Figure 4.5 can be viewed as an instance of the convex-concave procedure [Yuille and Rangarajan, 2002].

[54] In our algorithm, the parameters $\mathbf{w}$ are updated by sub-gradient descent using the passive-aggressive update [Crammer et al., 2006]. The approximation made here is more plausible if the weight vectors do not change much, which can be achieved by setting a small learning rate.

APPENDIX B: DERIVATION OF PASSIVE-AGGRESSIVE UPDATE USED FOR OPTIMIZING EXPECTED AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC

In this chapter, we describe the derivation of the passive-aggressive update [Crammer et al., 2006] used in the algorithm that appears in Figure 4.5. We use the same notation that appears in Figure 4.5. In particular, we assume that the weights $\mathbf{w}_t$ from the previous epoch have already been computed. Similarly, we compute the most likely segmentations $\overline{\mathbf{s}}_i^+ = \text{argmax}_{\overline{\mathbf{s}} \sim (s_i^+, e_i^+)} \mathbf{w}_t \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}})$. We now derive the online passive aggressive update that is used to compute $\mathbf{u}_{j+1}$ given a new training pair $(\overline{v}_i, \overline{\mathbf{x}}_i^+, \overline{\mathbf{x}}_i^-, \overline{\mathbf{s}}_i^+)$.

B.1 Online Passive-Aggressive Update

The online passive-aggressive update is computed as the solution to the following optimization problem:

$$\mathbf{u}_{j+1} = \underset{\mathbf{u}}{\text{argmin}} \frac{1}{2} \|\mathbf{u} - \mathbf{u}_j\|^2 + C\xi \tag{B.1}$$

where,

$$\xi \geq 0 \text{ and } \xi \geq \left[1 - \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right]_+ \tag{B.2}$$

Intuitively, Equations B.1 and B.2 can be interpreted as simultaneously requiring the updated weight vector $\mathbf{u}_{j+1}$ to have a low loss on the i-th training example while not allowing the weight vector to deviate too much from the current estimate $\mathbf{u}_j$. The term ξ is a slack variable, and the parameter $C > 0$ controls the relative importance of ensuring that the

new weight vector has a low loss on the i-th example. The optimization problem in Equation B.1 is a convex optimization problem and can be solved analytically using techniques from convex analysis [Boyd and Vandenberghe, 2004].

The first observation we make is that if the hinge loss of the current example with respect to the current vector is zero (i.e., $\left[1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right]_+ = 0$), then clearly the solution of the optimization problem is given by $\mathbf{u}_{j+1} = \mathbf{u}_j$. Consider the other case where, $\left[1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right]_+ > 0$. In this case, we define the Lagrangian of the optimization problem by introducing Lagrange multipliers $\tau, \lambda \geq 0$,

$$\mathcal{L}(\mathbf{u}, \xi, \tau, \lambda) = \frac{1}{2}\|\mathbf{u} - \mathbf{u}_j\|^2 + C\xi + \tau\left(1 - \xi - \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right) - \lambda\xi \tag{B.3}$$

$$= \frac{1}{2}\|\mathbf{u} - \mathbf{u}_j\|^2 + \xi(C - \tau - \lambda) + \tau\left(1 - \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right) \tag{B.4}$$

We now minimize the Lagrangian with respect to the primal variables ($\mathbf{u}$ and ξ) amd maximize with respect to the dual variables (τ and λ), which is equivalent to satisfying the Karush-Kuhn-Tucker conditions [Boyd and Vandenberghe, 2004]. Setting the (sub) gradient of the Lagrangian in Equation B.4 with respect to $\mathbf{u}$ to be equal to zero, we can write:

$$\mathbf{u} = \mathbf{u}_j + \tau\left(\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-(\mathbf{u}_{j+1}))\right) \tag{B.5}$$

$$\approx \mathbf{u}_j + \tau\left(\boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-(\mathbf{u}_j))\right) \tag{B.6}$$

$$= \mathbf{u}_j + \tau\Delta\boldsymbol{\phi}_i \tag{B.7}$$

where $\overline{\mathbf{s}}_i^-(\mathbf{u}) = \text{argmax}_{\overline{\mathbf{s}}} \mathbf{u} \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})$ and we have used the notation $\overline{\mathbf{s}}_i^- = \overline{\mathbf{s}}_i^-(\mathbf{u}_j)$ and $\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-)$.[55] Thus, our remaining task is to determine the learning rate τ to be used in Equation B.7.

[55]The approximation in Equation B.6 is necessitated by the fact that it is not possible to compute $\overline{\mathbf{s}}_i^-(\mathbf{u}_{j+1})$ directly.

Now, notice that if $C - \tau - \lambda \neq 0$, then the Lagrangian can be made to approach $-\infty$. Since we need to maximize with respect to the dual variables, we reject this case and impose the following constraint on the dual variables:

$$C - \tau - \lambda = 0 \tag{B.8}$$

Further, since $\lambda \geq 0$, we conclude $\tau \leq C$. Finally, we substitute Equation B.7 back into the original Lagrangian in Equation B.4:

$$\begin{aligned}
\mathcal{L}(\mathbf{u}, \xi, \tau, \lambda) &= \frac{1}{2}\tau^2\|\Delta\boldsymbol{\phi}_i\|^2 + \tau\Big\{1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) \\
&\quad + \max_{\overline{\mathbf{s}}} \left\{\mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}) + \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\}\Big\} && \text{(B.9)} \\
&\approx \frac{1}{2}\tau^2\|\Delta\boldsymbol{\phi}_i\|^2 + \tau\Big\{1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) \\
&\quad + \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-) + \max_{\overline{\mathbf{s}}} \left\{\tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\}\Big\} && \text{(B.10)} \\
&\approx \frac{1}{2}\tau^2\|\Delta\boldsymbol{\phi}_i\|^2 + \tau\Big\{1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) - \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) \\
&\quad + \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-) + \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-)\Big\} && \text{(B.11)} \\
&= -\frac{1}{2}\tau^2\|\Delta\boldsymbol{\phi}_i\|^2 + \tau(1 - \mathbf{u}_j \cdot \Delta\boldsymbol{\phi}_i) && \text{(B.12)}
\end{aligned}$$

where in Equation B.10 we approximate $\max_{\overline{\mathbf{s}}} \left\{\mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}) + \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\} \approx \max_{\overline{\mathbf{s}}} \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}) + \max_{\overline{\mathbf{s}}} \tau\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})$ and in Equation B.11, we make the approximation that $\max_{\overline{\mathbf{s}}} \left\{\Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right\} \approx \Delta\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}}_i^-)$.[56] Notice that Equation B.12 is a quadratic equation in τ since it is of the form $(A\tau^2 + B\tau)$, which we are trying to maximize with respect to τ. Also, the leading coefficient $A = -\frac{1}{2}\|\Delta\boldsymbol{\phi}_i\|^2 < 0$, which implies that the maximum value of $A\tau^2 + B\tau$ occurs when $\tau = -\frac{B}{2A}$. Thus the value

[56] In principle, it is possible to compute the best segmentation without making this approximation by setting the weight vector to $\Delta\boldsymbol{\phi}_i$, and then computing the highest scoring segmentation. However, this would require an additional decode, which can be avoided by making the approximation.

of τ which maximizes the Lagrangian is given by,

$$\tau = \frac{(1 - \mathbf{u}_j \cdot \Delta\boldsymbol{\phi}_i)}{\|\Delta\boldsymbol{\phi}_i\|^2} \tag{B.13}$$

Substituting Equation B.13 into Equation B.7 and from Equation B.8 we can write,[57]

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_i \Delta\boldsymbol{\phi}_i \tag{B.14}$$

where,

$$\alpha_i = \min\left\{C, \frac{[1 - \mathbf{u}_j \cdot \Delta\boldsymbol{\phi}_i]_+}{\|\Delta\boldsymbol{\phi}_i\|^2}\right\} \tag{B.15}$$

The update derived in Equation B.15 corresponds to the update that we used in the algorithm that appears in Figure 4.5 with $C = \lambda^{-1}$.

[57] We write a single expression for both cases when $\left[1 - \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^+, \overline{v}_i, \overline{\mathbf{s}}_i^+) + \max_{\overline{\mathbf{s}}} \mathbf{u}_j \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i^-, \overline{v}_i, \overline{\mathbf{s}})\right]_+ = 0$ as well as when the hinge loss is greater than 0.

APPENDIX C: ARPABET PHONEMIC SYMBOLS

In Table C.1, we present a list of Arpabet phonemic symbols used in the Switchboard pronunciation dictionary, along with example transcriptions of the canonical pronunciations of words containing those symbols.[58]

Phoneme Symbol	IPA Symbol	Example Word	Phonemic Transcription
aa	ɑ	l**o**ck	/l aa k/
ae	æ	b**a**t	/b ae t/
ah	ʌ	b**u**t	/b ah t/
ao	ɔ	b**ou**ght	/b ao t/
aw	aʊ	c**ow**	/k aw/
ax	ə	**a**bout	/ax b aw t/
ay	aɪ	b**uy**	/b ay/
b	b	**b**et	/b eh t/
ch	tʃ	**ch**ur**ch**	/ch er ch/
d	d	**d**ebt	/d eh t/
dh	ð	**th**at	/dh ae t/
eh	ɛ	b**e**t	/b eh t/
el	əl	batt**le**	/b ae t el/
en	n̩	butt**on**	/b ah t en/
er	ɚ	b**ir**d	/b er d/
ey	eɪ	b**ai**t	/b ey t/
f	f	**f**at	/f ae t/
g	ɡ	**g**et	/g eh t/
hh	h	**h**ello	/hh ax l ow/
ih	ɪ	b**i**ts	/b ih t s/
iy	i	b**ea**t	/b iy t/
jh	dʒ	**j**udge	/jh ah jh/
k	k	**k**it	/k ih t/

continued on next page . . .

[58]The table is adapted from http://www.isip.piconepress.com/projects/switchboard/doc/education/phone_comparisons/ .

Table C.1 – continued from previous page

Phoneme Symbol	IPA Symbol	Example Word	Phonemic Transcription
l	l	**l**et	/l eh t/
m	m	**m**et	/m eh t/
n	n	**n**et	/n eh t/
ng	ŋ	si**ng**	/s ih ng/
ow	oʊ	b**oa**t	/b ow t/
oy	ɔɪ	b**oy**	/b oy/
p	p	**p**et	/p eh t/
r	ɹ	**r**ent	/r eh n t/
s	s	**s**at	/s ae t/
sh	ʃ	**sh**ut	/sh ah t/
t	t	**t**en	/t eh n/
th	θ	**th**ree	/th r iy/
uh	ʊ	b**oo**k	/b uh k/
uw	u	t**oo**	/t uw/
v	v	**v**at	/v ae t/
w	w	**w**it	/w ih t/
y	j	**y**ou	/y uw/
z	z	**z**oo	/z uw/
zh	ʒ	plea**s**ure	/p l eh zh er/

Table C.1: List of Arpabet phonemic symbols along with examples of words whose canonical pronunciations contain those symbols. For reference, the corresponding IPA symbols are also provided.

APPENDIX D: CANTONESE PHONE TO FEATURE MAPPING

In this chapter, we present a list of Cantonese phonemes along with their corresponding IPA symbols and articulatory mappings. For completeness, we also include a list of each of the articulatory feature streams with the set of possible articulatory feature values that these streams can take.[59] These articulatory feature streams represent the position and degree of constriction of the lips (LIP-LOC and LIP-OPEN), the tongue tip (TT-LOC and TT-OPEN), the tongue body (TB-OPEN and TB-LOC), and the state of the velum (VEL) and the glottis (GLOT). In order to represent the Cantonese phonemes in terms of their correspoding articulatory feature mappings, we use articulatory feature values as defined by Livescu [see Livescu, 2005, Appendix B] introducing a new symbol (ASP) for indicating the state of aspiration of the glottis. For completeness, we list the set of articulatory feature values (reproduced with modifications where appropriate, from [Livescu, 2005]) for the Cantonese phonemes used in our experiments in Table D.1.

In Table D.2, we list the mapping from each of the Cantonese phones to the corresponding AF values. Also indicated for reference is the mapping from SAMPA to corresponding IPA symbols. Note that following [Livescu, 2005], dipthongs, stops and affricates are separated into two symbols (e.g., the dipthong iw (IPA: iw is represented as two symbols iw_p1 and iw_p2 representing the first and second part of the dipthong respectively).) The

[59]The six Cantonese tone categories are not included in this list. In our experiments, we treat these as an additional aspect of the Glottis label.

Articulatory Feature	Description	Number of Feature Values	Feature Value = Meaning
LIP-LOC	position (roughly, horizontal displacement of the lips)	3	PRO = protruded (rounded) LAB = labial (default/neutral position) DEN = dental (labio-dental position)
LIP-OPEN	degree of opening of the lips	4	CL = closed CR = critical (labial/ labio-dental fricative) NA = narrow WI = wide (all other sounds)
TT-LOC	location of the tongue tip	3	DEN = interdental ALV = alveolar P-A = palato-alveolar
TT-OPEN	degree of opening of the tongue tip	5	CL = closed CR = critical M-N = medium-narrow MID = medium WI = wide
TB-LOC	location of the tongue body	4	PAL = palatal VEL = velar UVU = uvular (default/neutral position) PHA = pharyngeal
TB-OPEN	degree of opening of the tongue body	6	CL = closed (stop consonant) CR = critical NA = narrow M-N = medium-narrow MID = medium WI = wide
VEL	state of the velum	2	CL = closed (non-nasal) OP = open (nasal)
GLOT	state of the glottis	4	CL = closed (glottal stop) CR = critical (voiced) ***ASP = aspirated*** OP = open (voiceless)

Table D.1: Values for articulatory feature streams used in Cantonese STD experiments.

mapping was obtained by first mapping phonemic symbols to the corresponding phonological feature classes (e.g., place of articulation, manner of articulation etc.) (devised by Eric Fosler-Lussier, Yanzhang He, and Joo-Kyung Kim based on [Zee, 1999; Stokes et al., 2002; Wikipedia, 2013]). This mapping was converted into a phoneme to articulatory feature mapping as shown in Table D.2 by Karen Livescu.

Phoneme	IPA	LIP-LOC	LIP-OPEN	TT-LOC	TT-OPEN	TB-LOC	TB-OPEN	VEL	GLOT
_6	ɐ	LAB	WI	ALV	MID	UVU	MID	CL	CR
_6j_p1	ɐj	LAB	WI	ALV	MID	UVU	MID	CL	CR
_6j_p2	ɐj	LAB	WI	ALV	M-N	PAL	M-N	CL	CR
_6w_p1	ɐw	LAB	WI	ALV	MID	UVU	MID	CL	CR
_6w_p2	ɐw	PRO	NA	P-A	WI	UVU	M-N	CL	CR
_9:	œː	PRO	WI	ALV	MID	PAL	MID	CL	CR
_9y_p1	œj	PRO	WI	ALV	MID	PAL	MID	CL	CR
_9y_p2	œj	PRO	NA	ALV	M-N	PAL	NA	CL	CR
a:	aː	LAB	WI	ALV	WI	PHA	M-N	CL	CR
a:j_p1	aːj	LAB	WI	ALV	WI	PHA	M-N	CL	CR
a:j_p2	aːj	LAB	WI	ALV	M-N	PAL	M-N	CL	CR
a:w_p1	aːw	LAB	WI	ALV	WI	VEL	WI	CL	CR
a:w_p2	aːw	PRO	NA	P-A	WI	UVU	M-N	CL	CR
b_p1	b	LAB	CL	ALV	MID	UVU	WI	CL	OP
b_p2	b	LAB	CR	ALV	MID	UVU	WI	CL	OP
d_p1	d	LAB	WI	ALV	CL	VEL	MID	CL	OP
d_p2	d	LAB	WI	ALV	CR	VEL	MID	CL	OP
dz_p1	dz	LAB	WI	ALV	CL	VEL	MID	CL	OP
dz_p2	dz	LAB	WI	ALV	CR	UVU	MID	CL	OP
E:	ɛː	LAB	WI	ALV	MID	PAL	MID	CL	CR
ej_p1	ej	LAB	WI	ALV	MID	PAL	MID	CL	CR
ej_p2	ej	LAB	WI	ALV	M-N	PAL	M-N	CL	CR
f	f	DEN	CR	ALV	MID	VEL	MID	CL	OP
g_p1	g	LAB	WI	P-A	WI	VEL	CL	CL	OP
g_p2	g	LAB	WI	P-A	WI	VEL	CR	CL	OP
gw_p1	gw	PRO	NA	P-A	WI	VEL	CL	CL	OP
gw_p2	gw	PRO	NA	P-A	WI	VEL	CR	CL	OP
h	h	LAB	WI	ALV	MID	UVU	MID	CL	OP
i:	iː	LAB	WI	ALV	M-N	PAL	NA	CL	CR
iw_p1	iw	LAB	WI	ALV	M-N	PAL	NA	CL	CR
iw_p2	iw	PRO	NA	P-A	WI	UVU	NA	CL	CR
j	j	LAB	WI	ALV	M-N	PAL	NA	CL	CR
k_p1	k	LAB	WI	P-A	WI	VEL	CL	CL	ASP
k_p2	k	LAB	WI	P-A	WI	VEL	CR	CL	ASP
kw_p1	kw	PRO	NA	P-A	WI	VEL	CL	CL	ASP
kw_p2	kw	PRO	NA	P-A	WI	VEL	CR	CL	ASP
l	l	LAB	WI	ALV	CL	UVU	NA	CL	CR
m	m	LAB	CL	ALV	MID	UVU	MID	OP	CR
n	n	LAB	WI	ALV	CL	UVU	MID	OP	CR
N	ŋ	LAB	WI	P-A	WI	VEL	CL	OP	CR
O:	ɔː	PRO	WI	ALV	WI	PHA	M-N	CL	CR
O:j_p1	ɔːj	PRO	WI	ALV	WI	PHA	M-N	CL	CR
O:j_p2	ɔːj	LAB	WI	ALV	M-N	PAL	M-N	CL	CR
ow_p1	ow	PRO	WI	P-A	WI	UVU	M-N	CL	CR
ow_p2	ow	PRO	NA	P-A	WI	VEL	NA	CL	CR
p_p1	p	LAB	CL	ALV	MID	UVU	WI	CL	ASP

continued on next page . . .

Table D.2 – continued from previous page

Phoneme	IPA	LIP-LOC	LIP-OPEN	TT-LOC	TT-OPEN	TB-LOC	TB-OPEN	VEL	GLOT
p_p2	p	LAB	CR	ALV	MID	UVU	WI	CL	ASP
s	s	LAB	WI	ALV	CR	UVU	MID	CL	OP
t_p1	t	LAB	WI	ALV	CL	VEL	MID	CL	ASP
t_p2	t	LAB	WI	ALV	CR	VEL	MID	CL	ASP
ts_p1	ts	LAB	WI	ALV	CL	VEL	MID	CL	OP
ts_p2	ts	LAB	WI	ALV	CR	UVU	MID	CL	ASP
u:	u:	PRO	NA	P-A	WI	VEL	NA	CL	CR
u:j_p1	u:j	PRO	NA	P-A	WI	VEL	NA	CL	CR
u:j_p2	u:j	LAB	WI	ALV	M-N	PAL	M-N	CL	CR
w	w	PRO	NA	P-A	WI	UVU	NA	CL	CR
y:	y:	PRO	NA	ALV	M-N	PAL	NA	CL	CR
sil	-	DEN	CL	DEN	CL	PAL	CL	CL	CL

Table D.2: Mapping from Cantonese phones to corresponding articulatory feature values. The mapping from SAMPA symbols to the corresponding IPA symbols is adapted from http://www.phon.ucl.ac.uk/home/sampa/cantonese.htm.

BIBLIOGRAPHY

[IAR, 2011] "Babel program broad agency announcement IARPA-BAA-11-02," Available online: https://www.fbo.gov/index?id=78fece5f8ad57ddffb437dff607446fc, 2011.

[Agarwal, 2011] S. Agarwal, "The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list," in *Proceedings of the SIAM International Conference on Data Mining*, 2011.

[Akbacak et al., 2008] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[Atal et al., 1978] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique." *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1553, May 1978.

[Benzeghiba et al., 2007] M. Benzeghiba, R. D. Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: A review," *Speech Communication*, vol. 49, pp. 763–786, 2007.

[Bilmes and Zweig, 2002] J. Bilmes and G. Zweig, "The graphical models toolkit: An open source software system for speech and time-series processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2002, pp. 3916–3919.

[Boyd and Vandenberghe, 2004] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[Browman and Goldstein, 1992] C. P. Browman and L. Goldstein, "Articulatory phonology: an overview," *Phonetica*, vol. 49, no. 3–4, pp. 155–180, 1992.

[Browman and Goldstein, 1986] C. P. Browman and L. Goldstein, "Towards an articulatory phonology," *Phonology yearbook*, vol. 3, no. 21, pp. 219–252, 1986.

[Browman and Goldstein, 1990] C. P. Browman and L. Goldstein, "Tiers in articulatory phonology, with some implications for casual speech," in *Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech*, J. Kingston and M. E. Beckman, Eds. Cambridge University Press, 1990, pp. 341–376.

[Christiansen and Rushforth, 1977] R. W. Christiansen and C. K. Rushforth, "Detecting and locating key words in continuous speech using linear predictive coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 25, no. 5, pp. 361–367, 1977.

[Cortes and Mohri, 2004] C. Cortes and M. Mohri, "Confidence intervals for the area under the ROC curve," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2004.

[Crammer et al., 2006] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[Davis and Mermelstein, 1980] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[Dempster et al., 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[Deng et al., 1995] L. Deng, J. Nu, and H. Sameti, "Improved speech modeling and recognition using multi-dimensional articulatory states as primitive speech units," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1995, pp. 385–388.

[Deng et al., 1997] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Communication*, vol. 22, no. 2-3, pp. 93–111, 1997.

[Deng and Sun, 1994] L. Deng and D. X. Sun, "A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features," *Journal of the Acoustical Society of America*, vol. 95, no. 5, pp. 2702–2719, 1994.

[Erler and Freeman, 1996] K. Erler and G. H. Freeman, "An HMM-based speech recognizer using overlapping articulatory features," *Journal of the Acoustical Society of America*, vol. 100, no. 4, pp. 2500–2513, 1996.

[Farnetani and Recasens, 2012] E. Farnetani and D. Recasens, "Coarticulation and connected speech processes," in *Essential Clinical Skills, Volume 4 : The Handbook of Phonetic Sciences*, 2nd ed., W. J. Hardcastle, J. Laver, and F. E. Gibbon, Eds. Wiley-Blackwell, 2012, ch. 9, pp. 316–352.

[Fiscus et al., 2007] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proceedings of the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2007.

[Fosler et al., 1996] E. Fosler, M. Weintraub, S. Wegmann, Y.-H. Kao, S. Khudanpur, C. Galles, and M. Saraclar, "Automatic learning of word pronunciation from data," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1996.

[Fosler-Lussier et al., 2013] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional random fields in speech, audio, and language processing," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1054–1075, 2013.

[Frankel et al., 2000] J. Frankel, K. Richmond, S. King, and P. Taylor, "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," in *Proceedings of the IEEE International Conference on Spoken Language Processing (ICSLP)*, 2000.

[Frankel, 2003] J. Frankel, "Linear dynamic models for automatic speech recognition," Ph.D. dissertation, The Centre for Speech Technology Research, Edinburgh University, 2003.

[Frankel and King, 2001] J. Frankel and S. King, "ASR - articulatory speech recognition," in *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 2001, pp. 599–602.

[Frankel and King, 2005] J. Frankel and S. King, "A hybrid ANN/DBN approach to articulatory feature recognition," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, Lisbon, Portugal, 2005, pp. 3045–3048.

[Frankel et al., 2007a] J. Frankel, M. Magimai-Doss, S. King, K. Livescu, and Özgur Çetin, "Articulatory feature classifiers using 2000 hours of telephone speech," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2007.

[Frankel et al., 2007b] J. Frankel, M. Wester, and S. King, "Articulatory feature recognition using dynamic bayesian networks," *Computer Speech and Language*, vol. 21, no. 4, pp. 620–640, 2007.

[Gales, 1998] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[Garofolo et al., 1993] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus cdrom," NIST CD-ROM, 1993.

[Ghosh and Narayanan, 2010] P. K. Ghosh and S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *Journal of the Acoustical Society of America*, vol. 128, pp. 2162–2172, 2010.

[Ghosh and Narayanan, 2011] P. K. Ghosh and S. S. Narayanan, "A subject-independent acoustic-to-articulatory inversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[Giachin et al., 1990] E. Giachin, A. Rosenberg, and C.-H. Lee, "Word juncture modeling using phonological rules for HMM-based continuous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 1990, pp. 737–740.

[Godfrey et al., 1992] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1992, pp. 517–520.

[Greenberg et al., 1996] S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1996.

[Grézl et al., 2007] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 757–760.

[Hardcastle, 1985] W. Hardcastle, "Some phonetic and syntactic constraints on lingual coarticulation during /kl/ sequences," *Speech Communication*, vol. 4, no. 1–3, pp. 247–263, 1985.

[Hazen et al., 2009] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, 2009, pp. 421–426.

[Hermansky et al., 2000] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 2000, pp. 1635–1638.

[Hermansky, 1990] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990.

[Higgins and Wohlford, 1985] A. L. Higgins and R. E. Wohlford, "Keyword recognition using template concatenation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1985.

[Hinton et al., 2006] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554., 2006.

[Hiroya and Honda, 2004] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an HMM-based speech production model," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 175–185, march 2004.

[Hodgen et al., 1996] J. Hodgen, A. Lofqvist, V. Gracco, I. Zlokarnik, P. Rubin, and E. Saltzman, "Accurate recovery of articulator positions from acoustics: New conclusions based on human data," *Journal of the Acoustical Society of America*, vol. 100, no. 3, pp. 1819–1834, September 1996.

[Hu et al., 2010] C. Hu, X. Zhuang, and M. Hasegawa-Johnson, "FSM-based pronunciation modeling using articulatory phonological code," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2010.

[Hunter and Lange, 2004] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[Jansen and Durme, 2012] A. Jansen and B. V. Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proceedings of the annual confernece of the International Speech Communication Association (Interspeech)*, 2012.

[Jansen and Niyogi, 2009] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 8, pp. 1457–1470, 2009.

[Johnson et al., 2004] D. Johnson et al., "ICSI QuickNet software package," Available online:http://www.icsi.berkeley.edu/Speech/qn.html, 2004.

[Jou et al., 2006] S.-C. Jou, L. Maier-Hein, T. Schultz, and A. Waibel, "Articulatory feature classification using surface electromyography," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[Jurafsky et al., 2001] D. Jurafsky, W. Ward, Z. Jianping, K. Herold, Y. Xiuyang, and Z. Sen, "What kind of pronunciation variation is hard for triphones to model?" in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[Jyothi et al., 2012] P. Jyothi, E. Fosler-Lussier, and K. Livescu, "Discriminatively learning factorized finite state pronunciation models from dynamic Bayesian networks," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2012.

[Jyothi et al., 2011] P. Jyothi, K. Livescu, and E. Fosler-Lussier, "Lexical access experiments with context-dependent articulatory feature-based models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[Karanasou et al., 2012] P. Karanasou, L. Burget, D. Vergyri, M. Akbacak, and A. Mandal, "Discriminatively trained phoneme confusion model for keyword spotting," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2012.

[Keshet et al., 2009] J. Keshet, D. Grangier, and S. Bengio, "Discriminative keyword spotting," *Speech Communication*, vol. 51, pp. 317–329, 2009.

[King and Taylor, 2000] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 333–353, 2000.

[King et al., 2005] S. King, C. Bartels, and J. Bilmes, "SVitchboard 1: small vocabulary tasks from Switchboard," in *Proceedings of the annual conference of the Internation Speech Communication Association (Interspeech)*, Lisbon, Portugal, 2005, pp. 3385–3388.

[King et al., 2007] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[Kirchhoff, 1999] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. dissertation, University of Bielefeld, 1999.

[Kirchhoff et al., 2002] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, no. 3-4, pp. 303–319, 2002.

[Kschischang et al., 2001] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[Lafferty et al., 2001] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[Lee and Ellis, 2012] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proceedings of the annual conference of the International Speech Communiciation Association (Interspeech)*, 2012.

[Lee and Rose, 1998] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 49–60, January 1998.

[Leggetter and Woodland, 1995] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[Lippmann, 1997] R. P. Lippmann, "Speech recognition by machines and humans," *Speech Communication*, vol. 22, pp. 1–15, 1997.

[Livescu, 2005] K. Livescu, "Feature-based pronunciation modeling for automatic speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2005.

[Livescu et al., 2007b] K. Livescu, A. Bezman, N. Borges, L. Yung, O. Çetin, J. Frankel, S. King, M. Magimai-Doss, X. Chi, and L. Lavoie, "Manual transcription of conversational speech at the articulatory feature level," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[Livescu et al., 2007a] K. Livescu, O. Çetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magami-Doss, and K. Saenko, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 621–624.

[Livescu et al., 2012] K. Livescu, E. Fosler-Lussier, and F. Metze, "Subword modeling for automatic speech recognition: Past, present, and emerging approaches," *Signal Processing Magazine*, vol. 29, no. 6, pp. 44–57, 2012.

[Livescu and Glass, 2004a] K. Livescu and J. Glass, "Feature-based pronunciation modeling for speech recognition," in *Proceedings of the annual conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Stroudsburg, PA, USA, 2004, pp. 81–84.

[Livescu and Glass, 2004b] K. Livescu and J. Glass, "Feature-based pronunciation modeling with trainable asynchrony probabilities," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Jeju Island, Korea, 2004, pp. 677–680.

[Mangu et al., 2000] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[Manos and Zue, 1997] A. S. Manos and V. W. Zue, "A study on out-of-vocabulary word modeling for a segment-based keyword spotting system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1997.

[McAllaster et al., 1998] D. McAllaster, L. Gillick, F. Scattone, and M. Newman, "Fabricating conversational speech data with acoustic models : A program to examine model-data mismatch," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.

[McGowan, 1994] R. S. McGowan, "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests," *Speech Communication*, vol. 14, no. 1, pp. 19–48, 1994.

[Metze and Waibel, 2002] F. Metze and A. Waibel, "A flexible stream architecture for ASR using articulatory features," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, Colorado, USA, 2002, pp. 2133–2136.

[Miller et al., 2007] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2007.

[Miller and Nicely, 1955] G. A. Miller and P. E. Nicely, "An analysis of perceptual confusions among some english consonants," *Journal of the Acoustical Society of America*, vol. 27, pp. 338–352, 1955.

[Mitra et al., 2009] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman, and L. Goldstein, "Noise robustness of tract variables and their application to speech recognition," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2009.

[Mitra et al., 2011a] V. Mitra, H. Nam, and C. Y. Espy-Wilson, "Robust speech recognition using articulatory gestures in a dynamic Bayesian network framework," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[Mitra et al., 2011c] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Speech inversion: benefits of tract variables over pellet trajectories," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[Mitra et al., 2011b] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 1913–1924, 2011.

[Morgan and Bourlard, 1995] N. Morgan and H. Bourlard, "Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 24–42, 1995.

[Morris and Fosler-Lussier, 2008] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 3, pp. 617–628, march 2008.

[Morris, 2010] J. Morris, "A study on the use of conditional random fields for automatic speech recognition," Ph.D. dissertation, The Ohio State University, Department of Computer Science and Engineering, 2010.

[Murphy, 2002] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[Muscariello et al., 2011] A. Muscariello, G. Gravier, and F. Bimbot, "Zero-resource audio-only spoken term detection based on a combination of template matching techniques," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2011.

[Næss et al., 2011] A. Næss, K. Livescu, and R. Prabhavalkar, "Articulatory feature classification using nearest neighbors," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2011.

[Narayanan et al., 2011] S. Narayanan, E. Bresch, P. K. Ghosh, L. Goldstein, A. Katsamanis, Y. Kim, A. Lammert, M. Proctor, V. Ramanarayanan, and Y. Zhu, "A multimodal real-time MRI articulatory corpus for speech research," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2011, pp. 837–840.

[Norouzian et al., 2013] A. Norouzian, R. Rose, S. Hamidi, and A. Jansen, "Zero resource graph-based confidence estimation for open vocabulary spoken term detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[Ostendorf, 1999] M. Ostendorf, "Moving beyond the 'beads-on-a-string' model of speech," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 1999.

[Pallett et al., 1999] D. S. Pallett, J. G. Fiscus, J. S. Garofolo, A. Martin, and M. Przybocki, "1998 broadcast news benchmark test results: English and non-english word error rate performance measures," in *DARPA Broadcast News Workshop*, 1999.

[Papcun et al., 1992] G. Papcun, J. Hochberg, T. R. Thomas, F. Laroche, J. Zacks, and S. Levy, "Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data," *Journal of The Acoustical Society of America*, vol. 92, no. 2, pp. 688–700, 1992.

[Plahl et al., 2009] C. Plahl, B. Hoffmeister, G. Heigold, J. Lööf, R. Schlüter, and H. Ney, "Development of the GALE 2008 mandarin LVCSR system," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2009.

[Povey and Woodland, 2002] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. 105–108.

[Prabhavalkar et al., 2011] R. Prabhavalkar, E. Fosler-Lussier, and K. Livescu, "A factored conditional random field model for articulatory feature forced transcription," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[Prabhavalkar et al., 2012] R. Prabhavalkar, J. Keshet, K. Livescu, and E. Fosler-Lussier, "Discriminative spoken term detection with limited data," in *Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, 2012, available online: http://www.ttic.edu/sigml/symposium2012/papers/prabhavalkar_mlslp2012.pdf.

[Prabhavalkar et al., 2013] R. Prabhavalkar, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative articulatory models for spoken term detection in low-resource conversational settings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[Rabiner and Juang, 1993] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall Inc., 1993.

[Rakotomamonjy, 2012] A. Rakotomamonjy, "Sparse support vector infinite push," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

[Richardson et al., 2003] M. Richardson, J. Bilmes, and C. Diorio, "Hidden-articulator Markov models for speech recognition," *Speech Communication*, vol. 41, pp. 511–529, 2003.

[Richmond, 2001] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, University of Edinburgh, 2001.

[Riley et al., 1999] M. Riley, W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, and G. Zavaliagkos, "Stochastic pronunciation modelling from hand-labelled phonetic corpora," *Speech Communication*, vol. 29, pp. 209–224, 1999.

[Rohlicek et al., 1989] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1989, pp. 627–630.

[Rose et al., 1996] R. C. Rose, J. Schroeter, and M. M. Sondhi, "The potential role of speech production models in automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 99, no. 3, pp. 1699–1709, 1996.

[Rose and Paul, 1990] R. Rose and D. Paul, "A hidden Markov model based keyword recognition system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1990, pp. 129–132.

[Rudin, 2009] C. Rudin, "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list," *Journal of Machine Learning Research*, vol. 10, pp. 2233–2271, 2009.

[Saltzman and Munhall, 1989] E. L. Saltzman and K. G. Munhall, "A dynamical approach to gestural patterning in speech production," *Ecological Psychology*, vol. 1, no. 4, pp. 333–382, 1989.

[Saraçlar and Khudanpur, 2004] M. Saraçlar and S. Khudanpur, "Pronunciation change in conversational speech and its implications for automatic speech recognition," *Computer Speech and Language*, vol. 18, no. 4, pp. 375–395, 2004.

[Saraçlar et al., 2000] M. Saraçlar, H. Nock, and S. Khudanpur, "Pronunciation modeling by sharing gaussian densitites across phonetic models," *Computer Speech and Language*, vol. 14, no. 2, pp. 137–160, 2000.

[Schroeter and Sondhi, 1994] J. Schroeter and M. M. Sondhi, "Techniques for estimating vocal-tract shapes from the speech signal," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 133–150, jan 1994.

[Silaghi and Bourlard, 1999] M.-C. Silaghi and H. Bourlard, "Iterative posterior-based keyword spotting without filler models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.

[Stephenson et al., 2000] T. A. Stephenson, H. Bourlard, S. Bengio, and A. C. Morris, "Automatic speech recognition using dynamic bayesian networks with both acoustic and articulatory variables," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.

[Stokes et al., 2002] S. Stokes, J. T.-K. Lau, and V. Ciocca, "The interaction of ambient frequency and feature complexity in the diphthong errors of children with phonological disorders." *Journal of Speech, Language, and Hearing Research*, vol. 45, no. 6, pp. 1188–1201., 2002.

[Strik and Cucchiarini, 1999] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for asr: A survey of the literature," *Speech Communication*, vol. 29, no. 2-4, pp. 225–246, 1999.

[Stüker et al., 2003a] S. Stüker, F. Metze, T. Schultz, and A. Waibel, "Integrating multilingual articulatory features into speech recognition," in *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 2003.

[Stüker et al., 2003b] S. Stüker, T. Schultz, F. Metze, and A. Waibel, "Multilingual articulatory features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.

[Sun et al., 2000] J. Sun, X. Jing, and L. Deng, "Data-driven model construction for continuous speech recognition using overlapping acoustic features," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, 2000.

[Sutton, 2006] C. Sutton, "GRMM: GRaphical Models in Mallet," Available online: http://mallet.cs.umass.edu/grmm/, 2006.

[Sutton and McCallum, 2012] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[Sutton et al., 2004] C. Sutton, K. Rohanimanesh, and A. McCallum, "Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

[Suzuki et al., 1998] S. Suzuki, T. Okadome, and M. Honda, "Determination of articulatory positions from speech acoustics by applying dynamic articulatory constraints," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.

[Szöke et al., 2005] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, and J. Černockỳ, "Phoneme based acoustics keyword spotting in informal continuous speech," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, V. E Matoušek, P. Mautner, and T. Pavelka, Eds. Springer Berlin Heidelberg, 2005, vol. 3658, pp. 302–309.

[Tajchman et al., 1995] G. Tajchman, E. Fosler, and D. Jurafsky, "Building multiple pronunciation models for novel words using exploratory computational phonology," in *Proceedings of Eurospeech*, 1995, pp. 2247–2250.

[Tang et al., 2012] H. Tang, J. Keshet, and K. Livescu, "Discriminative pronunciation modeling: A large-margin, feature-rich approach," in *Proceedings of the Association of Computational Linguistics (ACL)*, 2012.

[Vergyri et al., 2007] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2007.

[Wallace et al., 2011] R. Wallace, B. Baker, R. Vogt, and S. Sridharan, "Discriminative optimisation of the figure of merit for phonetic spoken term detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 1677–1687, 2011.

[Weide, 2007] R. Weide, "The Carnegie Mellon pronouncing dictionary [cmudict. 0.7a]," Available online: http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 2007.

[Weintraub et al., 1996] M. Weintraub, K. Taussig, K. Hunicke-smith, and A. Snodgrass, "Effect of speaking style on LVCSR performance," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1996, pp. 16–19.

[Westbury, 1994] J. R. Westbury, *X-ray microbeam speech production database user's handbook, Version 1.0*, Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, WI, USA, June 1994.

[Wester, 2003] M. Wester, "Pronunciation modeling for ASR - knowledge-based and data-derived methods," *Computer Speech and Language*, vol. 17, no. 1, pp. 69–85, 2003.

[Wikipedia, 2013] Wikipedia, "Cantonese phonology — wikipedia, the free encyclopedia," Available online: http://en.wikipedia.org/wiki/Cantonese_phonology, 2013, [Online; accessed 17-June-2013].

[Wilpon et al., 1990] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[Wrench, 2001] A. Wrench, "A new resource for production modeling in speech technology," in *Workshop on Innovations in Speech Processing*, Stratford-upon-Avon, UK, 2001.

[Wrench and Richmond, 2000] A. A. Wrench and K. Richmond, "Continuous speech recognition using articulatory data," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.

[Young et al., 2002] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge University Press, 2002.

[Yuille and Rangarajan, 2002] A. Yuille and A. Rangarajan, "The convex-concave computational procedure (CCCP)," in *Proceedings of Neural Information Processsing Systems (NIPS)*, 2002.

[Zee, 1999] E. Zee, "Chinese (Hong Kong Cantonese)," in *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge U.K.: Cambridge University Press, 1999, pp. 58–60.

[Zhang and Glass, 2009] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[Zhuang et al., 2008] X. Zhuang, H. Nam, M. Hasegawa-Johnson, L. Goldstein, and E. Saltzman, "The entropy of articulatory phonological code: Recognizing gestures from tract variables," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2008.

[Zhuang et al., 2009] X. Zhuang, H. Nam, M. Hasegawa-Johnson, E. Saltzman, and L. Goldstein, "Articulatory phonological code for word classification," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, 2009.

[Zweig, 1998] G. Zweig, "Speech recognition with dynamic bayesian networks," Ph.D. dissertation, University of California at Berkeley, 1998.

[Zweig and Nguyen, 2010] G. Zweig and P. Nguyen, "SCARF: A segmental conditional random field toolkit for speech recognition," in *Proceedings of the annual conference of the International Speech Communication Association (Interspeech)*, Makuhari, Japan, 2010, pp. 2858–2861.