Automatic Dialect and Accent Recognition and its Application to Speech Recognition

Fadi Biadsy

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2011

ABSTRACT

Automatic Dialect and Accent Recognition and its Application to Speech Recognition

Fadi Biadsy

A fundamental challenge for current research on speech science and technology is understanding and modeling individual variation in spoken language. Individuals have their own speaking styles, depending on many factors, such as their dialect and accent as well as their socioeconomic background. These individual differences typically introduce modeling difficulties for large-scale speaker-independent systems designed to process input from any variant of a given language. This dissertation focuses on automatically identifying the dialect or accent of a speaker given a sample of their speech, and demonstrates how such a technology can be employed to improve Automatic Speech Recognition (ASR).

In this thesis, we describe a variety of approaches that make use of multiple streams of information in the acoustic signal to build a system that recognizes the regional dialect and accent of a speaker. In particular, we examine frame-based acoustic, phonetic, and phonotactic features, as well as high-level prosodic features, comparing generative and discriminative modeling techniques. We first analyze the effectiveness of approaches to language identification that have been successfully employed by that community, applying them here to dialect identification. We next show how we can improve upon these techniques. Finally, we introduce several novel modeling approaches – Discriminative Phonotactics and kernel-based methods. We test our best performing approach on four broad Arabic dialects, ten Arabic sub-dialects, American English vs. Indian English accents, American English Southern vs. Non-Southern, American dialects at the state level plus Canada, and three Portuguese dialects.

Our experiments demonstrate that our novel approach, which relies on the hypothesis that certain phones are realized differently across dialects, achieves new state-of-the-art performance on most dialect recognition tasks. This approach achieves an Equal Error Rate (EER) of 4% for four broad Arabic dialects, an EER of 6.3% for American vs. Indian English accents, 14.6% for American English Southern vs. Non-Southern dialects, and 7.9% for three Portuguese dialects. Our framework can also be used to automatically extract linguistic knowledge, specifically the context-dependent phonetic cues that may distinguish one dialect form another. We illustrate the efficacy of our approach by demonstrating the correlation of our results with geographical proximity of the various dialects.

As a final measure of the utility of our studies, we also show that, it is possible to improve ASR. Employing our dialect identification system prior to ASR to identify the Levantine Arabic dialect in mixed speech of a variety of dialects allows us to optimize the engine's language model and use Levantine-specific acoustic models where appropriate. This procedure improves the Word Error Rate (WER) for Levantine by 4.6% absolute; 9.3% relative. In addition, we demonstrate in this thesis that, using a linguistically-motivated pronunciation modeling approach, we can improve the WER of a state-of-the art ASR system by 2.2% absolute and 11.5% relative WER on Modern Standard Arabic.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Completing a PhD thesis is impossible without years of gracious help from many colleagues. I have been blessed with a network of friends who have shown an abundance of excitement and passion during my five years at Columbia. I would like to thank them all for being there to listen to me when I was excited about something and for being interested in my work.

First and foremost I would like to thank my committee (Julia Hirschberg, Kathy McKeown, Nizar Habash, Dan Ellis, and Michael Collins) for their excellent guidance throughout my years of study. I had been told at the start that the key to a successful and joyful PhD experience is finding a good adviser. Julia has been one of the best things that has happened to me. I have been very lucky to work with her all these years. I have learned a great deal from her, from speech processing and how scientific experiments should be conducted to professional skills. Her door has always been open to me to talk about research questions or about anything else for which I needed advice. I could not have gone this far without her help, encouragement, passion, inspiration and friendship. I would like to thank Kathy McKeown for teaching me that one can be extremely intelligent and down to earth at the same time. I am exceedingly thankful to her for being a friend in addition to an adviser, for her guidance, and for believing in me. I am extremely thankful to Nizar Habash for his support and encouragement throughout my academic career. He has been always available when I needed his help. I would also like to thank him for his useful discussions and feedback, and for his contribution to this work. I am very thankful to Dan Ellis for answering my questions and for suggesting ideas in my proposal that made this work more successful. I would also like to express my gratitude to Michael Collins for being available for discussions and most importantly for his collaboration on the kernel-based approach which plays an important role in this work.

I would like to thank Lidia Mangu for serving on my PhD proposal committee, for being always interested in my research, and for her constant helpful feedback. She made my summer internships at IBM T.J. Watson very successful and enjoyable. I also have much gratitude for the rest of the IBM speech team especially, Hong-Kwang Kuo, Jiri Navratil, Jason Pelecanos, Michael Picheny, George Saon, and Hagen Soltau for very helpful and informative discussions. I am very thankful to Daisy Nguyen who gave me access to many machines at Columbia to run my experiments. I would like to thank Pedro Torres-Carrasquillo and Nancy Chen for providing me with the file segmentation definitions which allowed direct comparison to their work. I am also thankful to Dimitra Vergyri for sharing with me the details of the SRI Arabic pronunciation dictionary. I would like to thank Alberto Abad and Isabel Trancoso for providing me with the acoustic features of their Portuguese data.

To my best friends Wisam Dakka and Gabriel Zaccak who have always been there to encourage and support me for more than ten years in any phase in my life, I am deeply thankful and happy to have such friends. Their friendships are invaluable. My gratitude also goes to Bob Coyne who has been such a great friend who listened and helped me pass any obstacle during my PhD. We have always had challenging and interesting debates. I am also very thankful to my friend and officemate David Elson who made my stay at work enjoyable all these years, for useful discussion, and for his constant help finding the best words to describe my work. I am very happy to have met Kapil Thadani who has been fun to debate with about research problems, although only after long debates are we able to agree on their solutions. I am thankful for the support of my friend Andrew Rosenberg with whom I spent a great and fun time with during my PhD. I also would like to thank Erica Cooper and Daniel Bauer for their interest in my work. I am also thankful to, Mohamed Altantawy, Hila Becker, Naseem Biadsy, Sasha Caskey, Tarek Dakka, Mona Diab, Frank Enos, Laura Furst, Agustin Gravano, Luis Gravano, Kevin Lerman, Jackson Liscombe, Sameer Maskey, Michael Mandel, Azzam Mari, Helena Moniz, Kristen Parton, Owen Rambow, Sara Stolbach, Volker Dellwo, and William Yang Wang for being invaluable colleagues and friends.

Last but not least, all my gratitude goes to my family and to Jennifer, who has been there to support and encourage me all these years. She makes my life so much more enjoyable. Without her, this program would have been very hard.

To my mother who inspired me to dream and be kind.

To my father who taught me to worship education.

Chapter 1

Introduction

A fundamental challenge for current research on speech science and technology is understanding and modeling individual variation in spoken language. Individuals have their own speaking styles, depending on many factors, including the dialect and accent of the speaker as well as the socioeconomic background of the speaker and contextual variables such as the degree of familiarity between the speaker and hearer and the register of the speaking situation, from very casual to very formal [Eskenazi, 1992].

The past few decades have seen considerable progress in automatically identifying the language of a speaker given a sample of his/her speech. Accent and dialect recognition have more recently begun to receive attention from the speech science and technology communities. The task of dialect identification is the recognition of a speaker's regional dialect, within a predetermined language, given the acoustic signal alone. Dialect recognition is a difficult problem in particular since even within the same accent/dialect or register individual variation may occur; for example, in spontaneous speech, some speakers tend to exhibit more articulation reduction (e.g., reducing or deletion of function words) than others. The problem of dialect recognition has been viewed as more challenging than that of language recognition due to the greater similarity between dialects of the same language. Although dialects may differ in any dimension(s) of the linguistic spectrum including, morphological, lexical, syntactic, phonetic and phonological differences, these differences are likely to be more subtle across dialects than those across languages.

Dialect identification helps in Automatic Speech Recognition (ASR) since speakers with different dialects pronounce some words differently, consistently altering certain phones and even morphemes. This is evident, for example, with the Arabic language, which has multiple variants, including Modern Standard Arabic (MSA), the formal written standard language of the media, culture and education, and the informal spoken dialects that are the preferred method of communication in daily life. While there are commercially available ASR systems for recognizing MSA with low error rates (typically trained on Broadcast News), these recognizers fail when a native Arabic speaker speaks in his/her regional dialect. Even in news broadcasts, speakers often *code switch* between MSA and dialect, especially in conversational speech, such as that found in interviews and talk shows. Being able to identify dialect vs. MSA as well as to identify which dialect is spoken prior to the recognition process allows for the use of a more restricted pronunciation dictionary in decoding, resulting in a reduced search space. Moreover, it will enable the ASR system to adapt its acoustic, morphological, and language models appropriately.

Identifying the regional dialect of a speaker will also provide important benefits for speech technology beyond improving speech recognition. It will allow us to infer the speaker's regional origin and ethnicity and to adapt features used in speaker recognition to regional origin. It should also prove useful in telephony-based help systems, either adapting the output of text-to-speech synthesis in a spoken dialogue system to produce regional speech or directing the telephone conversation to an agent whose dialect is the same as the caller. In addition, it can be helpful in forensic speaker profiling in judicial or military situations. Finally, Arabic dialect identification is helpful for identifying charismatic speakers. In our own work, we have observed that the more dialectal words a speaker utters, the less charismatic he/she is perceived [Biadsy *et al.*, 2007].

In this thesis, we describe different approaches and modeling techniques that make use of multiple streams of information in the acoustic signal to build a system that identifies the regional dialect of speakers. In particular, we examine frame-based acoustic, phonetic, phonotactic, and high-level prosodic features as well as modeling techniques for identifying four broad Arabic dialects (Levantine, Gulf, Iraqi, and Egyptian). We first analyze the effectiveness of some approaches that have been successfully employed by the language recognition community for the task of Arabic dialect recognition. Then, we improve upon these approaches as well as analyzing our novel methodology. Afterwards, we test our best performing approach on ten Arabic sub-dialects, American English vs. Indian English accents, American English Southern vs. Non-Southern, American dialects at the state level, and three Portuguese dialects. We also demonstrate how an Arabic dialect identification system can improve Arabic speech recognition.

Most of our approaches to Arabic dialect recognition rely heavily on MSA phone hypotheses. An essential component of ASR and phone recognition systems is the pronunciation dictionary (lexicon), which maps the orthographic representation of words to their phonetic or phonemic pronunciation variants. The correspondence between orthography and pronunciation in MSA falls somewhere between that of languages such as Spanish and Finnish, which have an almost one-to-one mapping between letters and sounds, and languages such as English and French, which exhibit a more complex letter-to-sound mapping [El-Imam, 2004]. The more complex this mapping is, the more difficult the language is for ASR. In this thesis, we also describe a method which relies on linguistically motivated pronunciation rules applied on the output of a morphological analyzer and disambiguation tool to automatically construct pronunciation dictionaries. We show that using such dictionaries improve phone and word recognition.

This thesis represents the following contributions: We design a new pronunciation modeling technique for MSA, relying on morphological analysis, that significantly improves phone recognition as well as a state-of-the-art Arabic ASR system. Using this pronunciation modeling approach, we examine the effectiveness of two well-known phonotactic-based approaches, Phone Recognition followed by Language Modeling (PRLM) and Parallel-PRLM, on Arabic dialects. We improve the standard speaker and language recognition acoustic-based approach, Gaussian Mixture Model-Universal Background Model (GMM-UBM), using feature space Maximum Likelihood Linear Regression transforms (fMLLR). We also propose a new approach and new features to model the prosodic structure of dialects and use these features to identify prosodic structure differences across four broad Arabic dialects. More importantly, we invent two novel techniques, Discriminative Phonotactics and Kernel-based phonetic methods, for dialect and accent recognition that yield our best results, with significant improvement over previous approaches. We also show that our framework can be used to automatically identify phonetic knowledge that contributes to our understanding of how dialects differ. Using the Kernel-based approach, we achieve a new state-of-the-art performance on most evaluated dialect and accent tasks. Another important contribution of this work is showing that Arabic dialect identification system can be used to improve ASR for Levantine Arabic. Finally, we show that we can summarize the phonetic content of any given utterance of any duration with a single vector of a fixed size, a representation that we hypothesize can benefit multiple speech processing technologies (e.g., speaker verification and identification).

This thesis is organized as follows. In Chapter 2, we explain some linguistic aspects of Arabic, primarily the pronunciation of MSA and phonological variations of Arabic dialects. Based on these pronunciation phenomena, we discuss, in Chapter 3, our new method to improve pronunciation modeling for ASR and phone recognition for MSA. In Chapter 4, we describe some related work to dialect recognition. This chapter also describes the general framework we adopt for dialect recognition. We also describe the corpora we employ for the four broad Arabic dialects. We analyze the performance of phonotactic approaches on these dialects in Chapter 5. We suggest a method to model the prosodic structure of dialects to improve a phonotactic approach in Chapter 6, and identify prosodic differences across these four Arabic dialects. We also test and improve upon a standard acoustic modeling-based approach in Chapter 7. We discuss our novel methods, discriminative phonotactics and our kernel-based approach, in Chapters 8 and 9, respectively. We then test our best-performing dialect-recognition system, the kernel-based approach, on dialects of languages other than Arabic in Chapter 10. Employing our Arabic dialect recognition system, we show how we can improve Arabic ASR system in Chapter 11. Finally, we conclude and propose directions of future work in Chapter 11.8.

Chapter 2

Arabic Linguistic Background

2.1 Introduction

The Arabic language has multiple variants, including Modern Standard Arabic (MSA), the formal written standard language of the media, culture and education across the Arab world. MSA is syntactically, morphologically and phonologically based on Classical Arabic, the language of the Qur'an (Islam's Holy Book). Lexically, however, it is much more modern. MSA is not a native language of any Arab.

The Arabic dialects, in contrast, are the true native language forms. They are generally restricted in use to informal daily communication. They are not taught in schools or even standardized, although there is a rich popular dialect culture of folktales, songs, movies, and TV shows. Dialects are primarily spoken, not written. However, this is changing as more Arabs gain access to electronic media such as emails and newsgroups. The Arabic dialects we see today originate from historical interactions between Classical Arabic and languages of the contemporaneous cultures. For example, Algerian Arabic has many influences from Berber as well as French. Arabic dialects differ substantially from MSA and each other in terms of phonology, morphology, lexical choice and syntax.

In this chapter, we explain some linguistic aspects of Arabic. These will motivate our approaches to MSA pronunciation modeling and dialect recognition. See [Habash, 2010] for additional computational and non-computational linguistic aspects of the Arabic language.

2.2 Arabic Orthography and Pronunciation

MSA is written in a morpho-phonemic orthographic representation using the *Arabic script*, an alphabet accented with optional diacritical marks (see [Biadsy *et al.*, 2006; Habash, 2010] for the details of the Arabic script).[1] MSA has 34 phonemes (28 consonants, 3 long vowels and 3 short vowels). The Arabic script has 36 basic letters (ignoring ligatures) and 9 diacritics. Most Arabic letters have a one-to-one mapping to an MSA phoneme; however, there is a small number of common exceptions [El-Imam, 2004; Habash *et al.*, 2007] which we summarize next.

2.2.1 Optional Diacritics

Arabic script commonly uses nine optional diacritics: (a) three short-vowel diacritics representing the vowels /a/, /u/ and /i/; (b) one long-vowel diacritic (Dagger Alif ‘) representing the long vowel /A/ in a small number of words; (c) three *nunation* diacritics (*F* /an/, *N* /un/, *K* /in/) representing a combination of a short vowel and the nominal indefiniteness marker /n/ in MSA; (d) one consonant lengthening diacritic (called Shadda ∼) which repeats/elongates the previous consonant (e.g., *kat∼ab* is pronounced /kattab/); and (e) one diacritic on consonants for marking when there is no diacritic (called Sukun *o*) – i.e., an indication that the consonant does not precede a short vowel.

Arabic diacritics can only appear *after* a letter. Word-initial diacritics (in practice, only short vowels) are handled by adding an extra Alif ا *A* (also called Hamzat-Wasl) at the beginning of the word. Sentence/utterance initial Hamzat-Wasl is pronounced like a glottal stop preceding the short vowel; however, the sentence medial Hamzat-Wasl is silent except for the short vowel. For example, *Ainkataba kitAbN* is /Ginkataba kitAbun/ but *kitAbN Ainkataba* is /kitAbun inkataba/. A ‘real’ Hamza (glottal stop) is always pronounced as a glottal stop. The Hamzat-Wasl appears most commonly as the Alif of the definite article *Al*. It also appears in specific words and word classes such as relative pronouns (e.g., *Al*y*

[1] In this thesis, we provide Arabic script orthographic transliteration in the Buckwalter transliteration scheme [Buckwalter, 2004]. For MSA phonetic symbols, we use a variant of the Buckwalter transliteration with the following exceptions: glottal stops are represented as /G/ and long vowels as /A/, /U/ and /I/. All Arabic script diacritics are phonologically spelled out.

'who').

Arabic short vowel diacritics are used together with the glide consonant letters *w* and *y* to denote the long vowels /U/ (as *uw*) and /I/ (*iy*). This makes these two letters ambiguous in undiacritized transcripts.

Diacritics are largely restricted to religious texts and Arabic language school textbooks. In other texts, fewer than 1.5% of words contain a diacritic. Some diacritics are lexical (where word meaning varies) and others are inflectional (where nominal case or verbal mood varies). Inflectional diacritics are typically word final. Since nominal case, verbal mood and nunation have all disappeared in spoken dialectal Arabic, Arabic speakers do not always produce these inflections correctly or at all.

Much work has been done on automatic Arabic diacritization [Vergyri and Kirchhoff, 2004; Ananthakrishnan *et al.*, 2005; Zitouni *et al.*, 2006; Habash and Rambow, 2007]. In this thesis, we use the MADA (Morphological Analysis and Disambiguation for Arabic) system to diacritize Arabic [Habash and Rambow, 2005; Habash and Rambow, 2007]. MADA, which uses the Buckwalter Arabic morphological Analyzer databases [Buckwalter, 2004], provides the necessary information to determine Hamzat-Wasl through morphologically tagging the definite article; in most other cases it outputs the special symbol "{" for Hamzat-Wasl.

2.2.2 Hamza Spelling

The consonant Hamza (glottal stop /G/) has multiple forms in Arabic script: ء ', أ >, إ <, ؤ &, ئ }, آ |. There are complex rules for Hamza spelling that depend primarily upon its vocalic context. For example, } is used word medially and finally when preceded or followed by an /i/ vowel. Similarly, the Hamza form | is used when the Hamza is followed by the long vowel /A/.

Hamza spelling is further complicated by the fact that Arabic writers often replace hamzated letters with the un-hamzated form (أ > → ا *A*) or use a two-letter spelling, e.g.

ئ } → ء ى Y'. Due to this variation, the un-hamzated forms (particularly for أ > and إ <) are typically ignored in Arabic ASR evaluation. The MADA system regularizes most of these spelling variations as part of its analysis.

2.2.3 Morpho-phonemic Spelling

Arabic script includes a small number of morphemic/lexical phenomena, some very common:

- **Ta-Marbuta** The Ta-Marbuta (*p*) is typically a feminine ending. It appears word-finally, optionally followed by a diacritic. In MSA it is pronounced as /t/ when followed by a diacritic; otherwise it is silent. For example, *maktabapN* 'a library' is pronounced / maktabatun/.

- **Alif-Maqsura** The Alif-Maqsura (*Y*) is a silent derivational marker, which always follows a short vowel /a/ at the end of a word. For example, *rawaY* 'to tell a story' is pronounced /rawa/.

- **Definite Article** The Arabic definite article is a proclitic that assimilates to the first consonant in the noun it modifies if this consonant is alveolar or dental (except for *j*). These are the so-called Sun Letters: *t, v, d, *, r, z, s, $, S, D, T, Z, l,* and *n*. For example, the word *Al$ams* 'the sun' is pronounced /a$$ams/ not */al$ams/. The definite article does not assimilate to the other consonants, the Moon Letters. For example, the word Alqamar 'the moon' is pronounced /alqamar/ not */aqqamar/.

- **Silent Letters** A silent Alif appears in the morpheme +*uwA* /U/ which indicates masculine plural conjugation in verbs. Another silent Alif appears after some nunated nouns, e.g., kitaAbAF /kitAban/. In some poetic readings, this Alif can be produced as the long vowel /A/: /kitAbA/. Finally, a common irregular spelling is that of the proper name *Eamrw* /Eamr/ 'Amr' where the final w is silent.

2.3 Linguistic Aspects of Arabic Dialects

Arabic dialects vary on many dimensions, primarily, geography and social class. Geo-linguistically, the Arab world can be divided in many different ways. The following is only one of many classifications of the main Arabic dialects:

- **Gulf Arabic** (GLF) includes the dialects of Kuwait, Saudi Arabia, Bahrain, Qatar, United Arab Emirates, and Oman;
- **Iraqi Arabic** (IRQ) is the dialect of Iraq. In some dialect classifications, Iraqi Arabic is considered a sub-dialect of Gulf Arabic;
- **Levantine Arabic** (LEV) includes the dialects of Lebanon, Syria, Jordan, Palestine and Israel;
- **Egyptian Arabic** (EGY) covers the dialects of the Nile valley: Egypt and Sudan;
- **Maghrebi Arabic** covers the dialects of Morocco, Algeria, Tunisia and Mauritania. Libya is sometimes included;
- **Yeminite Arabic** is often considered its own class;
- **Maltese Arabic** is not always considered an Arabic dialect. It is the only Arabic variant that is considered a separate language and is written with Latin script.

Socially, it is common to distinguish three sub-dialects within each dialect region: city dwellers, peasants/farmers and Bedouins. The three degrees are often associated with a class hierarchy from rich, settled city-dwellers down to Bedouins.

The relationship between MSA and the dialect in a specific region is complex. Arabs do not think of these two as separate languages. This particular perception leads to a special kind of coexistence between the two forms of language that serve different purposes — a phenomenon linguists term *diglossia*. Although the two variants have clear domains of prevalence — formal written (MSA) versus informal spoken (dialect) — there is a large gray area in between which is often filled with a mixing of the two forms.

2.3.1 Phonological Variations among Arabic Dialects

Although Arabic dialects and MSA vary on many different levels — phonology, orthography, morphology, lexical choice and syntax — we will focus on phonological differences in this thesis. It is important to point out that since Arabic dialects are not standardized, their orthography may not always be consistent. However, this is not a relevant point to this thesis since we are interested in dialect recognition using acoustic data and without making use of dialectal transcripts. MSA’s phonological profile includes 28 consonants, three short vowels, three long vowels and two diphthongs (/ay/ and /aw/). Arabic dialects vary phonologically from standard Arabic and each other. Some of the common variations include the following [Holes, 2004; Habash, 2006]:

The MSA consonant (/q/) is realized as a glottal stop /’/ in Egy and Lev and as /g/ in Glf and Irq. For example, the MSA word /ṯari:q/ ‘road’ appears as /ṯari:’/ (Egy and Lev) and /ṯari:g/ (Glf and Irq). Other variants also are found in sub dialects such as /k/ in rural Palestinian (Lev) and /dj/ in some Glf dialects. These changes do not apply to modern and religious borrowings from MSA. For instance, the word for ‘Qur’an’ is never pronounced as anything but /qur’a:n/.

The MSA alveolar affricate (/dj/) is realized as /g/ in Egy, as /j/ in Lev and as /y/ in Glf. Irq preserves the MSA pronunciation. For example, the word for ‘handsome’ is /djami:l/ (MSA, Irq), /gami:l/ (Egy), /jami:l/ (Lev) and /yami:l/ (Glf).

The MSA consonant (/k/) is generally realized as /k/ in Arabic dialects with the exception of Glf, Irq and the Palestinian rural sub-dialect of Lev, which allow a /č/ pronunciation in certain contexts. For example, the word for ‘fish’ is /samak/ in MSA, Egy and most of Lev but /simač/ in Irq and Glf.

The MSA consonant /θ/ is pronounced as /t/ in Lev and Egy (or /s/ in more recent borrowings from MSA), e.g., the MSA word /θala:θa/ ‘three’ is pronounced /tala:ta/ in Egy and /tla:te/ in Lev. Irq and Glf generally preserve the MSA pronunciation.

The MSA consonant /δ/ is pronounced as /d/ in Lev and Egy (or /z/ in more recent borrowings from MSA), e.g., the word for ‘this’ is pronounced /ha:δa/ in MSA versus /ha:da/ (Lev) and /da/ Egy. Irq and Glf generally preserve the MSA pronunciation.

The MSA consonants /ḏ/ (emphatic/velarized d) and /$\underline{\delta}$/ (emphatic /δ/) are both normalized to /ḏ/ in Egy and Lev and to /$\underline{\delta}$/ in Glf and Irq. For example, the MSA sentence /$\underline{\delta}$alla yaḏrubu/ ‘he continued to hit’ is pronounced /ḏall yuḏrub/ (Lev) and /$\underline{\delta}$all yu$\underline{\delta}$rub/ (Glf). In modern borrowings from MSA, /$\underline{\delta}$/ is pronounced as /ẕ/ (emphatic z) in Egy and Lev. For instance, the word for ‘police officer’ is /$\underline{\delta}$a:biṯ/ in MSA but /ẕa:biṯ/

in Egy and Lev.

In some dialects, a loss of the emphatic feature of some MSA consonants occurs, e.g., the MSA word /laṯi:f/ ‘pleasant’ is pronounced as /lati:f/ in the Lebanese city sub-dialect of Lev. Emphasis typically spreads to neighboring vowels: if a vowel is preceded or succeeded directly by an emphatic consonant (/ḏ/, /s̱/, /ṯ/, /δ̱/) then the vowel becomes an emphatic vowel. As a result, the loss of the emphatic feature does not affect the consonants only, but also their neighboring vowels.

Other vocalic differences among MSA and the dialects include the following: First, short vowels change or are completely dropped, e.g., the MSA word /yaktubu/ ‘he writes’ is pronounced /yiktib/ (Egy and Irq) or /yoktob/ (Lev). Second, final and unstressed long vowels are shortened, e.g., the word /maṯa:ra:t/ ‘airports’ in MSA becomes /maṯara:t/ in many dialects. Third, the MSA diphthongs /aw/ and /ay/ have mostly become /o:/ and /e:/, respectively. These vocalic changes, particularly vowel drop, lead to different syllabic structures. MSA syllables are primarily light (CV, CV:, CVC) but can also be (CV:C and CVCC) in utterance-final positions. Egy syllables are the same as MSA’s although without the utterance-final restriction. Lev, Irq and Glf allow heavier syllables including word initial clusters such as CCV:C and CCVCC.

Chapter 3

Modern Standard Arabic Pronunciation Modeling

3.1 Introduction

Almost all the approaches we develop in this thesis for Arabic dialect recognition make use of an Arabic phone recognizer. An essential component for building a phone recognizer or a speech recognizer is the pronunciation dictionary (lexicon). It maps the orthographic representation of words to their phonetic or phonemic pronunciation variants (cf. the example in Figure 3.1).

For languages with complex letter-to-sound mappings, pronunciation dictionaries are typically written by hand. However, for morphologically rich languages, such as MSA, pronunciation dictionaries are difficult to create by hand, because of the large number of word forms, each of which has a large number of possible pronunciations. MSA words, for example, have fourteen features: part-of-speech, person, number, gender, voice, aspect, determiner proclitic, conjunctive proclitic, particle proclitic, pronominal enclitic, nominal case, nunation, idafa (possessed), and mood. MSA features are realized using both concatenative (affixes and stems) and templatic (root and pattern) morphology with a variety of morphological and phonological adjustments that appear in word orthography and interact with orthographic variations.

Fortunately, the relationship between orthography and pronunciation is relatively reg-

Word		Pronunciations
The	→	*/dh ah/* */dh iy/*
Read	→	*/r iy d/* */r eh d/*

Figure 3.1: Sample entries in an English pronunciation dictionary

ular and well understood for MSA. Also, recent automatic techniques for morphological analysis and disambiguation, such as MADA [Habash and Rambow, 2005; Habash and Rambow, 2007], can also be useful in automating part of the dictionary creation process. Nonetheless, most documented Arabic ASR systems appear to handle only a subset of Arabic phonetic phenomena; very few use morphological disambiguation tools. This typically results in graphemic-like acoustic models as opposed to phonetic. Modeling phones accurately is particularly important for our dialect recognition work. It has been shown that the more accurate the phone hypotheses, the more accurate the phonotactic-based language identification approach [Decker *et al.*, 2003].

In the next section, we briefly describe related work and the baseline systems (BasePR and BaseWR) employed in this chapter for the first evaluation. Afterwards, we describe the pronunciation rules we have developed based upon the linguistic phenomena described in Chapter 2. We then show how these rules are used, together with MADA, to automatically build pronunciation dictionaries for training and decoding. We then present evaluations of our phone- and word-recognition systems (XPR and XWR) on MSA, comparing these systems to the two baseline systems, BasePR and BaseWR. To ensure that our results will generalize when we use a state-of-the-art ASR system, we test the effect of an improved version of our approach on IBM's Attila Arabic ASR system [Soltau *et al.*, 2009].

3.2 Related Work

Most recent work on ASR for MSA uses a single pronunciation dictionary constructed by mapping each undiacritized word in the training transcripts to all of its diacritized Buckwalter analyses and/or to the diacritized versions of the word in the Arabic Treebank [Maamouri *et al.*, 2003; Afify *et al.*, 2005; Messaoudi *et al.*, 2006; Soltau *et al.*, 2007; Soltau *et al.*, 2009]. Each diacritized word is converted to a single pronunciation with a one-to-one mapping using some generally unspecified rules. None of these systems use morphological disambiguation to determine the most likely pronunciation of the word given its textual context.

Vergyri et al. [2008], in contrast, *do* use a morphological and disambiguation tool to predict word pronunciation based on context. They also apply some linguistically-motivated pronunciation rules (described in the next section) to convert the morphological analyses to pronunciations. They select the top choice from the MADA system for each word in the training transcript to train their system's acoustic models. For the dictionary used during decoding (decoding pronunciation dictionary), each undiacritized word in the training transcripts is mapped to the corresponding top MADA morphological analysis. In Evaluation I (Section 3.5), we train the acoustic models of our baseline phone recognition system (BasePR) using their ([Vergyri *et al.*, 2008]'s) method for generating the pronunciation dictionary. BaseWR is our baseline word recognition system trained using their pronunciation dictionary; and for during decoding, it employs their decoding pronunciation dictionary.

For the baseline for Evaluation II (Section 3.6), we use the pronunciation dictionary employed in the IBM *vowelized* system [Soltau *et al.*, 2009] described in detail in Section 3.6.2. In a different version of the IBM system (e.g., [Saon *et al.*, 2010]), an "unvowelized" pronunciation dictionary is used. This dictionary is created by simply running Buckwalter's morphology analyzer on every word in the vocabulary. Then, every undiacratized word is mapped to all the morphological analyses. Afterwards, all the diacritics (including the 3 MSA short vowels, but not nunations) in these analyses are removed. Thus, the set of the remaining Buckwalter symbols represents the phonemic inventory of the system. For example, one of the entries in the dictionary for the undicritized word *ktAb* (given the analysis *kitAbuN*), is *"ktAb → /k t A b N/"*. On MSA, this system performs worse than the vowelized system of [Soltau *et al.*, 2009]. Note that while, to our knowledge, there is no comprehensive study for Arabic pronunciation modeling for ASR, El-Imam [2004] discusses MSA pronunciation rules for speech synthesis.

3.3 Pronunciation Rules

As noted in Section 2.3, diacritization alone does not predict actual pronunciation in MSA. In this section, we describe a set of rules based on MSA phonology which will extend a diacritized word to a set of possible pronunciations. It should be noted that even MSA-trained speakers, such as broadcast news anchors, may not follow the "proper" pronunciation according to Arabic syntax and phonology. So we attempt to accommodate these pronunciation variants in our pronunciation dictionary.

The following rules are applied on each diacritized word.[1] These rules are divided into four categories: (I) a shared set of rules used in all systems compared (BasePR, BaseWR, XPR and XWR);[2] (II) a set of rules in BasePR and BaseWR which we modified for XPR and XWR; (III) a first set of new rules devised for our systems XPR and XWR; and (IV) a second set of new rules that generate additional pronunciation variants. Below we indicate, for each rule, how many words in the training corpus (335,324 words) had their pronunciation affected by the rule. We also show an example for each rule after applying it independently of the other rules. Although most of the rules can be applied in any arbitrary order, applying them in the following order guarantees the correctness of the output.

I. Shared Pronunciation Rules

1. **Dagger Alif:** ‘ → /A/

   (e.g., h‘*A → hA*A) (This rule affected 1.8% of all the words in our training data)

2. **Madda:** | → /G A/

   (e.g., Al|n → AlGAn) (affected 1.9%)

3. **Nunation:** AF → /a n/, F → /a n/, /K/ → /i n/, N → /u n/

   (e.g., kutubAF → kutuban) (affected 9.7%)

[1] The script that generates the pronunciation dictionaries from MADA output can be downloaded from *www.cs.columbia.edu/speech/software.cgi*.

[2] We have attempted to replicate the baseline pronunciation rules for [Vergyri *et al.*, 2008] based on published work and personal communications with the authors. Note that none of these rules are implemented in the IBM baseline system [Soltau *et al.*, 2009] (see Section 3.6).

4. **Hamza:** All Hamza forms: $', \}, \&, <, > \rightarrow$ /G/
   (e.g., >kala → Gakala) (affected 21.3%)

5. **Ta-Marbuta:** p → /t/
   (e.g., madrasapa → madrasata) (affected 15.3%)

II. Modified Pronunciation Rules

1. **Alif-Maqsura:** Y → /a/
   (e.g., salomY → saloma) (affected 4.2%) *(Baseline: Y → /A/)*

2. **Shadda:** Shadda is always removed
   (e.g., ba$∼ara → ba$ara) (affected 23.8%) *(Baseline: the consonant was doubled)*

3. **U and I:** uwo → /U/, iyo → /I/
   (e.g., makotuwob → makotUb) (affected 25.07%) *(Baseline: same rule but it inaccurately interacted with the baseline Shadda rule)*

III. New Pronunciation Rules

1. **Waw Al-jamaa:** suffixes uwoA → /U/
   (e.g., katabuwoA → katabU) (affected 0.4%)

2. **Sun letters:** if the definite article (Al) is followed by a sun letter, remove the *l* and replace A by /a/.
   (e.g., Al$amsu → a$amsu) (affected 8.1%)

3. **Definite Article:** Al → /a l/ (if tagged as Al+ by MADA)
   (e.g., wAlkitAba → walkitAba) (affected 30.0%)

4. **Hamzat-Wasl:** { is always removed.
   (affected 3.0%)

5. **"Al" in relative pronouns:** Al → /a l/
   (affected 1.3%)

IV. New Pronunciation Rules Generating Additional Variants

- **Ta-Marbuta:** if a word ends with Ta-Marbuta (p) followed by any diacritic, remove the Ta-Marbuta and its diacritic. Apply the rules above (I-III) on the modified word and add the output pronunciation.
  (e.g., marbwTapF → marbwTa) (affected 15.3%)

- **Case/Mood ending:** if a word ends with a short vowel (a, u, i), remove the short vowel. Apply rules (I-III) on the modified word, and add the output pronunciation (e.g., yaktubu → yaktub (affected 60.9%)

As a post-processing step in all systems, we remove the Sukun diacritic and convert every letter X to phoneme /X/. In XPR and XWR, we also remove short vowels that precede or succeed a long vowel.

3.4 Building the Pronunciation Dictionaries

As noted above, pronunciation dictionaries map words to one or more phonetically expressed pronunciation variants. These dictionaries are used for training and decoding in ASR systems. Typically, most data available to train large vocabulary ASR systems is orthographically (not phonetically) transcribed. There are two well-known alternatives for training acoustic models in ASR: (1) bootstrap training, when some phonetically annotated data is available, and (2) flat-start, when such data is not available [Young *et al.*, 2006]. In flat-start training, the pronunciation dictionary is used to map the orthographic transcription of the training data to a sequence of phonetic labels to train the initial monophone models. Next, the dictionary is employed again to produce networks of possible pronunciations which can be used in forced alignment to obtain the most likely phone sequence that matches the acoustic data. Finally, the monophone acoustic models are re-estimated. In our work, we refer to this dictionary as the **training pronunciation dictionary**. The second usage of the pronunciation dictionary is to generate the pronunciation models while decoding. We refer to this dictionary as the **decoding pronunciation dictionary**.

For languages like English, no distinction between decoding and training pronunciation dictionaries is necessary, since word pronunciations typically do not change based on context. Also, as noted in Section 2.3, short vowels and other diacritic markers are typically not orthographically represented in MSA texts. Thus ASR systems typically do not output fully diacritized transcripts. Diacritization is generally not necessary to make the transcript readable by Arabic-literate readers. Therefore, entries in the decoding pronunciation dictionary consist of undiacritized words that are mapped to a set of phonetically-represented diacritizations. However, every entry in the training pronunciation dictionary is a fully diacritized word mapped to a set of possible context-dependent pronunciations. (Recall that a word may get different diacritic markers (e.g., short vowels) based on its context/grammatical function.) Particularly in the training step, contextual information for each word is available from the transcript; so, for our work, we can use the MADA morphological tagger to obtain the most likely diacritics. As a result, the speech signal is mapped to a more accurate representation of the training transcript, which we hypothesize will lead to a better estimation of the acoustic-model parameters.

As noted in Section 3.1, pronunciation dictionaries for ASR systems are usually written by hand. However, Arabic's morphological richness makes it difficult to create a pronunciation dictionary by hand since there are a very large number of word forms, each of which has a large number of possible pronunciations. The relatively regular relationship between orthography and pronunciation and tools for morphological analysis and disambiguation such as MADA, however, make it possible to create such dictionaries automatically with some success.[3]

3.4.1 Training Pronunciation Dictionary

In this section, we describe an automatic approach to building a pronunciation dictionary for MSA that covers all words in the orthographic transcripts of the training data. First, for each utterance transcript, we run MADA to disambiguate each word based on its context in the transcript. MADA outputs all possible fully-diacritized morphological analyses for each word, ranked by its confidence, the MADA confidence score.[4] We thus obtain a

[3] The MADA system [Habash and Rambow, 2005; Habash and Rambow, 2007] reports 4.8% diacritic error rate (DER) on all diacritics and 2.2% (DER) when ignoring the last (inflectional) diacritic.

[4] In our training data, only about 1% of all words are not diacritized because of lack of coverage in the morphological analysis component.

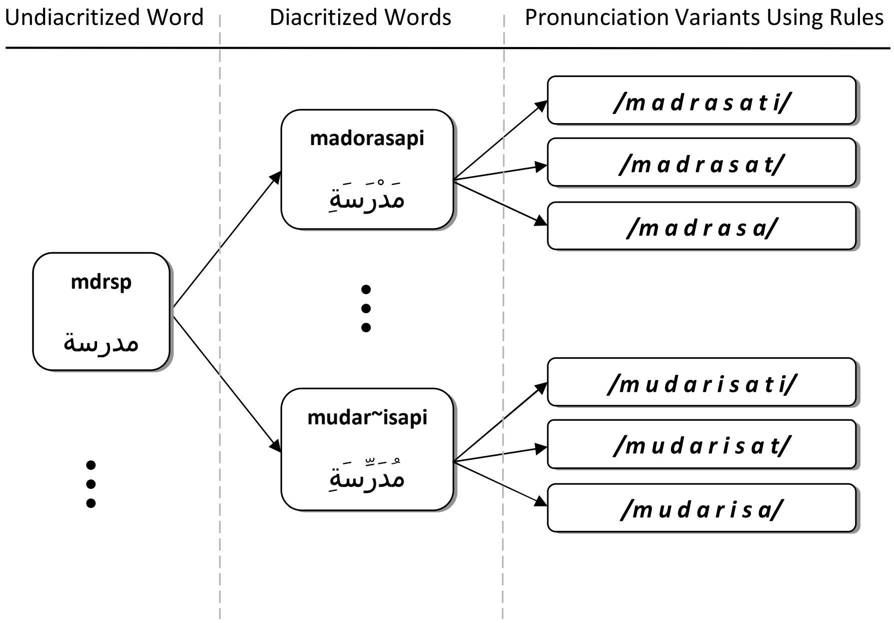


Figure 3.2: Mapping an undiacritized word to MADA outputs to possible pronunciations

fully-diacritized orthographic transcription for training. Second, we map the highest-ranked diacritization of each word to a set of pronunciations, which we obtain from the pronunciation rules described in Section 3.3. In Figure 3.2, the training pronunciation dictionary maps the 2^{nd} column (the entry keys) to the 3^{rd} column.

We generate the baseline training pronunciation dictionary using only the baseline rules from Section 3.3. This dictionary also makes use of MADA, but it maps the MADA-diacritized word to only one pronunciation. In other words, the baseline training dictionary maps the 2^{nd} column (the entry keys) to **only** one pronunciation in the 3^{rd} column in Figure 3.2.

3.4.2 Decoding Pronunciation Dictionary

The decoding pronunciation dictionary is used in ASR to build the pronunciation models used in decoding. Since, as noted above, it is standard to produce unvocalized transcripts when recognizing MSA, we must map word pronunciations to unvocalized orthographic output. Therefore, we normalize each diacritized word in our training pronunciation dictionary by removing diacritic markers and replace Hamzat-Wasl ({), <, and > by the letter 'A', and then map the modified word to the set of pronunciations for that word. For example, in

Figure 3.2, the undiacritized word *mdrsp* in the 1^{st} column is mapped to the pronunciations in the 3^{rd} column. Note that there is another advantage of generating pronunciations from the top MADA choice only over generating them from all the Buckwalter morphological analyses. MADA can also be viewed as a filter of uncommon or even errorful morphological analyses. In other words, if a word is never ranked as first in any context, it will not end up in the dictionary, thus resulting in a smaller and more precise dictionary. The baseline decoding pronunciation dictionary is constructed similarly but from the baseline training pronunciation dictionary.

3.5 Evaluation I

To determine whether our pronunciation rules are useful in speech processing applications, we evaluated their impact on two tasks, automatic phone recognition and ASR. For our experiments, we used the broadcast news TDT4 corpus (Arabic Set 1), divided, randomly, into 47.61 hours of speech (89 news shows) for training and 5.18 hours (11 shows) for testing.[5] Both training and test data were segmented based on silence and non-speech segments and down-sampled to 8Khz.[6] This segmentation produced 20,707 speech segments for our training data and 2,255 segments for testing.

3.5.1 Acoustic Models

Our monophone acoustic models are built using 3-state continuous Hidden Markov Models (HMMs) without state-skipping with a mixture of 12 Gaussians per state. We extract standard MFCC (Mel Frequency Cepstral Coefficients) features from 25 ms frames, with a frame shift of 10 ms. Each feature vector is 39D: 13 features (12 cepstral features plus energy), 13 deltas, and 13 double-deltas. The features are normalized using cepstral mean normalization. For our phone recognizer, the acoustic models are context-independent (i.e., monophone acoustic models are trained). For our ASR experiments, tied context-dependent

[5] http://projects.ldc.upenn.edu/TDT4/

[6] We down-sample because our ultimate goal is to use this system to decode telephone conversations for our Arabic dialect recognition approaches.

Gold Variants			*Open-loop (Accuracy)*		*Bigram Phone LM (Accuracy)*	
GV	*Acoustic Model of*	*Pron. Dict. of*	BasePR	XPR	BasePR	XPR
1	BasePR	BasePR	37.40	39.21	41.56	45.17
2	BasePR	BasePR+XPR	38.64	42.41	43.44	50.73
3	XPR	XPR	37.06	42.38	42.21	51.41
4	XPR	BasePR+XPR	37.47	42.74	42.59	51.51

Table 3.1: Comparing the effect of BasePR and XPR pronunciation rules, alone and in combination, using 4 Gold Variants under two conditions (Open-loop and LM)

cross-word triphone acoustic models are created with the same settings as monophones. The acoustic models are speaker- and gender-independent, Maximum Likelihood (ML)- trained from flat-start. We build our framework using the HMM Toolkit (HTK) [Young *et al.*, 2006].[7]

3.5.2 Phone Recognition Evaluation

We hypothesize that improved pronunciation rules will have a profound impact on phone recognition accuracy. To compare our phone recognition (XPR) system with the baseline (BasePR), we train two phone recognizers using HTK. The BasePR recognizer uses the training-pronunciation dictionary generated using the baseline rules; the XPR system uses a pronunciation dictionary generated using these rules plus our modified and new rules (cf. Section 3.4). The two systems are identical except for their pronunciation dictionaries.

We evaluate the two systems under two conditions: (1) phone recognition with a bigram phone language model (LM)[8] and (2) phone recognition with an open-loop phone recognizer, such that any phoneme can follow any other phoneme with a uniform distribution. Results of this evaluation are presented in Table 3.1.

Ideally, we would like to compare the performance of these systems against a common

[7] For Evaluation I, we have not employed advanced acoustic modeling techniques, such as vocal track length normalization, discriminative features, and speaker adaptation.

[8] The bigram phoneme LM of each phone recognizer is trained on the phonemes obtained from forced aligning the training transcript to the speech data using that recognizer's training pronunciation dictionary and acoustic models.

MSA phonetically-transcribed gold standard. Unfortunately, to our knowledge, such a data set does not exist. So we approximate such a gold standard on a blind test set through forced alignment, using the trained acoustic models and pronunciation dictionaries. Since our choice of acoustic model (of BasePR or XPR) and pronunciation dictionary (again of BasePR or XPR) can bias our results, we consider four *gold* variants (GV) with different combinations of acoustic model and pronunciation dictionary, to set expected lower and upper bounds.[9] These combinations are represented in Table 3.1 as GV1–4, where the source for the acoustic models is BasePR or XPR and source of pronunciation rules are BasePR, XPR or XPR and BasePR combined. These GV are described in more detail below, as we describe our results.

Since the BasePR system uses a pronunciation dictionary with a one-to-one mapping of orthography to phones, the GV1 phone sequence for any test utterance's orthographic transcript according to BasePR can be obtained directly from the orthographic transcript. Note that if, in fact, GV1 does represent the true gold standard (i.e., the correct phone sequence for the test utterances) then if XPR obtains a lower phone error rate using this gold standard than BasePR does, we can conclude that in fact XPR's acoustic models are better estimated. This is in fact the case. In the first line of Table 3.1, we see that XPR under both conditions (open-loop and bigram LM) significantly (p-value < 0.001) outperforms the corresponding BasePR phone recognizer using GV1.[10]

If GV1 does *not* accurately represent the phone sequences of the test data, then there must be some phones in the GV1 sequences that should be deleted, inserted, or substituted. On the hypothesis that our training-pronunciation dictionary might improve the BasePR assignments, we enrich the baseline pronunciation dictionary with XPR's dictionary. Now, we force-align the orthographic transcript using this extended pronunciation dictionary, still using BasePR's acoustic models, with the acoustic signal. We denote the output phone sequences as GV2. If a pronunciation generated using the BasePR dictionary was

[9] Note that the notion 'gold variant' here does not refer to human annotated phonetic labels. They are phonetic labels obtained form forced alignment.

[10] Throughout this discussion we use paired t-tests to measure significant difference, where the sample values are the phone recognizer accuracies on the utterances.

already correct (in GV1) according to the acoustic signal, this forced alignment process still has the option of choosing it. We hypothesize that the result, GV2, is a more accurate representation of the true phone sequences in the test data, since it should be able to model the acoustic signal more accurately. On GV2, as on GV1, we see that XPR, under both conditions, significantly (p-value < 0.001) outperforms the corresponding BasePR phone recognizers (see Table 3.1, second line).

We also compared the performance of the two systems using upper bound variants. For GV3 we used the forced alignment of the orthographic transcription using only XPR's pronuncation dictionary with XPR's acoustic models. In GV4 we combine the pronunciation dictionary of XPR with BasePR dictionary and use XPR's acoustic models. Unsurprisingly, we find that the XPR recognizer significantly (p-value < 0.001) outperforms BasePR when using these two variants under both conditions (see Table 3.1, third and fourth lines).

The results presented in Table 3.1 compare the robustness of the acoustic models as well as the pronunciation components of the two systems. We also want to evaluate the accuracy of our pronunciation predictions in representing the actual acoustic signal. One way to do this is to see how often the forced alignment process choose phone sequences using the BasePR pronunciation dictionary as opposed to XPR's. We force align the test transcript — using the XPR acoustic models and only the XPR pronunciation dictionary — with the acoustic signal. We then compare the output sequences to the output of the forced alignment process where the **combined** pronunciations from BasePR+XPR and the XPR acoustic models were used. We find that the difference between the two is only 1.03% (with 246,376 phones, 557 deletions, 1696 substitutions, and 277 insertions). Thus, adding the BasePR rules to XPR does not appear to contribute a great deal to the representation chosen by forced alignment. In a similar experiment, we use the BasePR acoustic models instead of the XPR models and compare the results of using BasePR-pronunciation dictionary with the combination of XPR+BasePR's dictionaries for forced alignment. Interestingly, in this experiment, we *do* find a significantly larger difference between the two outputs 17.04% (with 233,787 phones, 1404 deletions, 14013 substitutions, and 27040 insertions). We can conclude from these experiments that the baseline pronunciation dictionary alone is not sufficient to represent the acoustic signal accurately, since large numbers of phonemes are edited when adding the XPR pronunciations. In contrast, adding the BasePR's pronunciation dictionary to XPR's shows a relatively small percentage of edits, which suggests that the XPR pronunciation dictionary extends and covers more accurately the pronunciations already contained in the BasePR dictionary.

3.5.3 Speech Recognition Evaluation

We have also conducted an ASR experiment to evaluate the usefulness of our pronunciation rules for this application. We employ the baseline pronunciation rules to generate the baseline training and decoding pronunciation dictionaries. Using these dictionaries, we build the baseline ASR system (BaseWR). Using our extended pronunciation rules, we generate our dictionaries and train our ASR system (XWR).

Both systems have the same model settings, as described in Section 3.5.1. Recall that the ASR systems employ context-dependent acoustic models (triphones). Both systems also share the same language model (LM), a trigram LM trained on the undiacritized transcripts of the training data and a subset of Arabic gigawords (approximately 281 million words, in total), using the SRILM toolkit [Stolcke, 2002]. Recall that the training and testing acoustic data were down-sampled to 8Khz which is well-known to significantly increase the WER. Table 3.2 presents the comparison of BaseWR to the XWR system.

To evaluate the impact of the set of rules that generate additional pronunciation variants (described in Section 3.3 - IV) on word recognition, we built a system, denoted as XWR_I-III, that uses only the first three sets of rules (I–III) and compared its performance to that of both BaseWR and XWR system. As shown in Table 3.2, we observe that XWR_I-III significantly outperforms BaseWR in 2.27% (absolute) (p-value < 0.001). Also, XWR that uses all the rules (including IV set) significantly outperforms XWR_I-III in 1.24% (p-value < 0.001).

The undiacritized vocabulary size used in our experiment was 34,511. We observed that 6.38% of the words in the test data were out of vocabulary (OOV), which may partly explain the high WER. We have done some error analysis to understand some of the reasons behind high error rates for both systems. We observed that many of the test utterances are very noisy. We wanted to see whether XWR still outperforms BaseWR if we remove these

System	**WER**	Corr (%)	Del (%)	Sub (%)	Ins (%)
BaseWR	47.22	65.36	0.98	33.7	12.6
XWR_I–III	44.95	66.84	0.88	32.3	11.8
XWR	43.71	69.06	0.75	30.2	12.7

Table 3.2: Comparing the performance of BaseWR, XWR I–III, and XWR **Corr**ect is accuracy without counting insertions (%); total number of words: 36,538

utterances. Removing all utterances for which BaseWR obtains an WER of more than 75%, we are left with 1720/2255 utterances. On these remaining utterances, the BaseWR WER is 35.6% and XWR's WER is 32.77% — a significant difference despite the bias in favor of BaseWR.

3.6 Evaluation II + Improvements

We have seen a significant reduction in WER using our pronunciation modeling approach for our HTK-based ASR system. In this section, we further evaluate the impact of our approach on a state-of-the-art Arabic ASR system – The IBM GALE Arabic ASR system [Soltau *et al.*, 2009]. The basic settings of the system are described next.

3.6.1 ASR System

In the IBM Arabic ASR system, the input speech is represented by 13-dimensional Perceptual Linear Prediction (PLP) features extracted from 25ms frames, with a frame-shift of 10ms, with cepstral mean and variance normalization. Each frame is spliced together with four preceding and four succeeding frames and Linear Discriminant Analysis (LDA) is performed to yield 40-dimensional feature vectors.

The phonetic/allophonics acoustic model topology is the same as our HTK system, 3-state left-to-right HMMs, without state-skipping, but with a 2-state HMM for each of the three Arabic short vowels (/a/, /i/, and /u/). All models in this system have penta-phone cross-word acoustic context, as opposed to triphone in the HTK system. The number of context-dependent states is 3000, with a total of 50,000 Gaussian components. The acoustic models were ML-trained on 50 hours of speech, randomly selected from the Arabic GALE BN collection. Unlike our HTK-based experiment, all acoustic data here are sampled with 16Khz.

The following speaker compensation transforms are computed using the most likely word sequence hypothesis from the first-pass of a speaker-independent system: (1) Vocal-tract length normalization (VTLN), (2) feature space Maximum Likelihood Linear Regression (fMLLR), followed by (3) Maximum Likelihood Linear Regression (MLLR).

The system utilizes a 4-gram language model (of about 880 million n-geams) resulted from interpolating 20 language models trained with modified Kneser-Ney smoothing [Kneser and Ney, 1995] from the following resources: Transcripts of the audio data, Arabic Gigaword corpus, and Web transcripts for broadcast conversations collected by CMU/ISL (28M words from Al-Jazeera). The interpolation weights were optimized using GALE eval07 of BN and BC data which comprises about 74K words.

3.6.2 Pronunciation Dictionary Baseline

Both pronunciation dictionaries (training and decoding) are created by mapping every undiacritized word in the training data (speech transcripts and language model data) to all possible Buckwalter morphological analyses, as described in [Soltau *et al.*, 2007; Soltau *et al.*, 2009; Saon *et al.*, 2010]. One pronunciation variant is created for every analysis, by simply representing every letter and diacritic as a phoneme – except for the shadda diacritic, for which the consonant is doubled. Thus, the phonetic inventory consists of 43 phonemes. We can see that this is almost a letter-to-sound mapping; in other words, the representation is almost graphemic as opposed to phonemic – almost no pronunciation rules (except for the shadda rule).

The training pronunciation dictionary has on average 3.4 pronunciations per word, and the decoding pronunciation dictionary has 3.3 pronunciations per word for a total of 774K vocabulary size without pronunciation probabilities. The training pronunciation dictionary contains pronunciations for only the audio transcript vocabulary, whereas the decoding pronunciation dictionary contains, in addition, pronunciations for the vocabulary of the language model training data. Although this mapping is very simplistic, various ASR parameter settings (such as acoustic weights and number of context-dependent acoustic model states) have been tuned for such a representation. It should be noted that the morphology analyzer ignores the context of each word, in contrast to using MADA which utilizes the context to disambiguate/rank the analyses, thus more accurate pronunciations.

3.6.3 Training Pronunciation Dictionary

The training pronunciation dictionary is built using the same methodology explained in Section 3.4.1 on the transcripts of the 50 hour training data. Since the ASR system we employ here make use of penta-phone acoustic context, we modified the shadda rule here (rule II.2 above), by simply doubling the consonant instead of removing the shadda (∼). For example: (e.g., ba\$∼ara → /b a \$ \$ a r a/). We hypothesize that the phonetic decision tree using penta-phone contexts can disambiguate whether consonants are lengthened or not.

Since MADA may not always rank the best analysis as its top choice, we also run the pronunciation rules on the **second** best choice returned by MADA, when the difference between the top two choices is less than a threshold determined empirically (in our implementation we chose 0.2). The IBM system is flexible enough to allow specifying multiple word options at the transcript level. A sentence can be a sequence of word pairs as opposed to a sequence of single words. Figure 3.3 illustrates an input utterance transcript in the first line. The second line is produced after running MADA and choosing the first and second MADA options for some of the words. The second line is the input to the ASR trainer. We test whether adding the second MADA choice to the training pronunciation dictionary/transcript improves WER or not. The pronunciation dictionary has an entry for each diacritized word in the second line (possibly, with multiple pronunciations using the pronunciation rules above). Our training pronunciation dictionary has 1.72 pronunciations per word in average.

3.6.4 Decoding Pronunciation Dictionary

We follow the same procedure described in Section 3.4.2 to build the decoding dictionary for this evaluation using the pronunciation rules except for the shadda rule. We run MADA

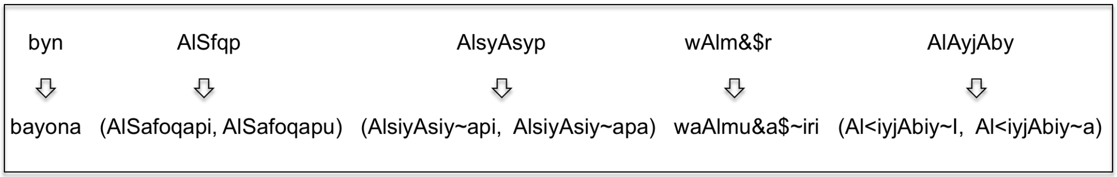


Figure 3.3: An illustration of the mapping of undiacritized words in the transcript to one or two diacritized words

on the transcripts of the speech training data and on the Arabic Giga Word corpus to build the decoding dictionary. Recall that in this dictionary, all pronunciations produced (by the pronunciation rules) for all diacritized word instances (from MADA first and second choices) of the same undiacritized and normalized form are mapped to this form. A pronunciation confidence score is calculated for each pronunciation. These scores are typically termed pronunciation probabilities, but in the IBM's system they need not form a probability distribution – scores do not have to sum to one.

We compute a pronunciation score s for a pronunciation p as the average of the MADA confidence scores of the MADA analyses of the word instances that this pronunciation was generated from. We compute this score for each pronunciation of a normalized undiacritized word. Let m be the maximum of these scores. Now, the final pronunciation confidence score for p is $-log_{10}(c/m)$. This basically means that the best pronunciation receives a penalty of 0 when chosen by the ASR decoder. This dictionary has about 3.6 pronunciations per word when using the first and second MADA choices.

3.6.5 Experiments

We build multiple ASR systems to evaluate the effectiveness of our pronunciation modeling using IBM's Attila framework. We test the following systems on GALE dev07 data set (2.6 hours of speech from 212 speakers; total 825 utterances; number of reference words is 18,186) and report their WER in Table 3.3.

- IBMBase: This is the baseline system which utilizes the baseline dictionaries described in Section 3.6.2.

- IBMXR-T1: This system employs the training pronunciation dictionary described in Section 3.6.3 to train the acoustic models. In this system, we include pronunciations of the first MADA choice only. The decoding pronunciation dictionary is the same as the one in the baseline system (IBMBase), but using our rules to produce pronunciations applied on Buckwalter's analyses. In other words, MADA is not employed for the decoding dictionary. Also this decoding dictionary lacks pronunciation confidence scores.

- IBMXR-T2: This system employs the same training dictionary as IBMXR-T1 but we include both the first and second MADA choices as pronunciations. The decoding pronunciation dictionary is exactly as in IBMXR-T1.

- IBMXR-T1-D2: This system employs the same training dictionary as IBMXR-T1 (first MADA choice). But now, the decoding pronunciation dictionary is created from MADA first and second choices as described in Section 3.6.4. Note that the dictionary has pronunciation confidence scores. In other words, MADA is used for both training and decoding pronunciation dictionaries.

- IBMXR-T2-D2: This system employs the same training dictionary as IBMXR-T1-D2 (first and second MADA choices). The decoding pronunciation dictionary is the same as in IBMXR-T2.

System	**WER**	Sub (%)	Del (%)	Ins (%)
IBMBase	19.2	13.4	4.3	1.5
IBMXR-T1	18.2	13.2	3.1	1.8
IBMXR-T2	18.1	13.1	3.2	1.9
IBMXR-T1-D2	17.4	12.5	3.2	1.7
IBMXR-T2-D2	17.0	12.2	3.2	1.6

Table 3.3: Comparing systems with different pronunciation dictionaries

We observe that all the systems that use our pronunciation dictionaries significantly outperform the baseline system (IBMBase). We obtain the most reduction in WER with

IBMXR-T2-D2, the system that uses the first and second MADA choices for the training and decoding pronunciation dictionaries. The difference between IBMXR-T1-D2 and IBMXR-T2-D2 is statistically significant (p-value=0.005). There is no significance difference between IBMXR-T1 and IBMXR-T2. This may suggest that the pronunciations generated from the MADA first choice are generally good enough to represent the actual spoken words. However, surprisingly, the difference between the performance of these two systems is signifiant when the MADA decoding dictionary is used in both (IBMXR-T1-D2, and IBMXR-T2-D2). The difference between the WER of every other pair of systems in Table 3.3 is statistically significant. Note that we have also experimented using the first 3 MADA choices in our dictionaries; nevertheless we have not obtained a reduction in WER. Moreover, obtaining significant improvement in IBMXR-T1 and IBMXR-T2-D2 suggests that the acoustic models are more robustly estimated ("sharper") than the baseline's. On the other hand, improving further the WER using IBMXR-T2-D2 suggests also that the pronunciation models (used at the ASR decoding stage) represent MSA pronunciations more accurately.

We conduct another experiment to test whether the gain we obtain over the baseline system holds when more than 50 hours of acoustic training data is used. We train the two systems IBMBase and IBMXR-T2-D2 using 1,500 hours of speech. We employ the same language model as the other experiments. Testing on the same test set (dev07), as shown in Table 3.4, IBMXR-T2-D2 significantly outperforms IBMBase in 1.7% absolute EER (11.1% relative) (p-value < 0.001).

System	**WER**	Sub (%)	Del (%)	Ins (%)
IBMBase	15.3	10.7	3.0	1.6
IBMXR-T2-D2	13.6	9.7	2.1	1.8

Table 3.4: Comparing systems when using 1,500 hours of training data

3.7 Conclusions

In this chapter, we have shown that the use of more linguistically motivated pronunciation rules can improve phone recognition and word recognition results for MSA. In particular, we run a morphological analysis and disambiguation tool (i.e., MADA) on the training data (acoustic transcripts and LM texts) to rank morphological analyses based on contexts. We then apply our pronunciation rules on the most likely analyses to generate two pronunciation dictionaries: one is used to train the ASR's acoustic models and the other is used during ASR decoding.

We have demonstrated that these dictionaries can significantly improve both MSA phone recognition and MSA WER. We have conducted a series of experiments to compare our HTK-based XPR and XWR systems to the corresponding baseline systems BasePR and BaseWR. We have obtained a significant improvement of 3.77%–7.29% (absolute) in phone error rate, and a significant improvement of 3.5% (absolute) in WER in the HTK-based ASR system. Testing our pronunciation modeling approach on a state-of-the-art Arabic ASR system, we have obtained 2.2% improvement in absolute WER (11.5% relative). We have found for this system that using the first and second MADA choices in the training and decoding dictionaries provide the best results. Also, we have seen how we could produce pronunciation confidence scores for the decoding pronunciation dictionary from MADA's confidence scores. Finally, our approach still significantly outperforms the baseline (11.1% relative reduction in WER) when far more acoustic training data is employed.

Chapter 4

Dialect Recognition

4.1 Introduction

A variety of cues by which humans and machines distinguish one language from another have been explored in previous research on language recognition. Such cue types include phone inventory and phonotactics, prosody, lexicon, morphology, and syntax. In this chapter, we discuss important related work employed in the language and dialect recognition community. We then explain the probabilistic framework adopted to guide us in exploring some of our baseline and novel approaches for dialect recognition. In this thesis, we first test these approaches on four broad Arabic dialects (described in Section 4.3). In Chapters 5–9 we present results of these Arabic experiments. We then, in Chapter 10, test the best performing approach (our Kernel-based method, described in Chapter 9) on Arabic sub-dialects and on dialects of languages other than Arabic.

4.2 Related Work

Some successful approaches to language identification have made use of phonotactic variation. For example, the parallel Phone Recognition followed by Language Modeling (parallel PRLM) [Zissman, 1996] approach uses phonotactic information – i.e., rules that govern phonemes and their sequences in a language – to identify languages using n-gram language models over phone sequence hypotheses. Zissman et al. [1996] show that the PRLM approach yields good results classifying Cuban and Peruvian dialects of Spanish in the Miami corpus, using an English phone recognizer. The recognition accuracy of this system on these two dialects is 84%, using up to 3 minutes of test utterances. Shen et al. [2008] describe a dialect recognition system that makes use of adapted phonetic models per dialect applied in a PRLM framework to distinguish American vs. Indian English and two Mandarin dialects (Mainland and Taiwanese). Using the adapted phonetic models outperforms the PRLM system in about 2% (absolute).

Gaussian Mixture Models - Universal Background model (GMM-UBM) has also achieved considerable success in speaker and language recognition [Reynolds *et al.*, 2000; Wong *et al.*, 2000]. Torres-Carrasquillo et al. [2004] develop a system using GMM-UBM with shifted delta cepstral (SDC) features. The system performs worse than that of [Zissman *et al.*, 1996] on the Miami corpus (70% accuracy) but performs well on two Mandarin dialects and two Spanish dialects from CallHome [Canavan and Zipperlen, 1996e].

Discriminative training has proven quite useful in recent language recognition systems (e.g., [Burget *et al.*, 2006; Matejka *et al.*, 2006]). Torres-Carrasquillo et al. [2008] show that GMM-UBM-based models – discriminatively trained with SDC features with an eigen-channel compensation component to remove language independent information, and vocal-tract length normalization (VTLN) – provide good results for the recognition of American vs. Indian English, four Chinese dialects, and three Arabic dialects (Gulf, Iraqi, and Levantine). Specifically, in this system, a GMM-UBM is initially ML-trained using the data from all dialects of the same language. A GMM for each dialect is then generated from the GMM-UBM by Maximum-A-Posteriori (MAP) adaptation. Following [Matejka *et al.*, 2006], they further discriminatively train the dialect GMMs with the Maximum Mutual Information (MMI) criterion, where the objective function is the posterior probability of correctly classifying all training utterances, using the extended Baum-Welch algorithm [Povey, 2004].

Alorfi explores ergodic HMMs to model phonetic differences between two Arabic dialects (Gulf and Egyptian Arabic) employing standard MFCC features [Alorfi, 2008]. Ma et al. [2006] use multi-dimensional pitch flux features and MFCC features to distinguish three Chinese dialects.[1] These pitch flux features reduce the error rate by more than 30% when

[1] The authors define pitch flux features for a given frame as the covariance of autocorrelation with its

added to a GMM based MFCC system. Given 15s of test-utterances, the system achieves an accuracy of 90% on the three dialects.

Recent approaches attempt to automatically extract linguistic (typically phonetic) rules that distinguish pairs of accents/dialects. Chen et al. [2010]'s system automatically identifies a set of biphones which discriminate American vs. Indian English accents using log-likelihood ratios. In particular, the authors compare the log-likelihood of the acoustic data of a given biphone under an adapted American English acoustic model to the log-likelihood under an adapted Indian English acoustic model. These set of biphones are subsequently used to adapt accent-dependent phone recognizer's acoustic models used for accent recognition. Their approach achieves an EER of 14.7%, and when fused with the PRLM approach, they obtain, similar to [Torres-Carrasquillo *et al.*, 2008] on the this task, an EER of 10.6%. Koller et al. [2010] build a system that makes use of acoustic data and orthographic transcripts to automatically identify a set monophones that distinguish pairs of Portuguese varieties (African, Brazilian, and European). For each pair of varieties, for each phone class, a binary Multi-Layer Perceptron classifier is trained to distinguish which variety a phone belongs to. The discriminating monophones are identified as those with the best performing classifiers. Using this knowledge, a phone recognizer is trained utilizng an augmented monophone inventory (standard Portuguese phones + the most discriminating phones). Employing this phone recognizer in a PRLM approach, the authors obtain a significant reduction in error over the Parallel-PRLM approach.

Intonational cues have been shown to be useful indicators for human subjects identifying regional dialects. Peters et al. [2002] show that human subjects rely on intonational cues to identify two German dialects (Hamburg urban dialects vs. Northern Standard German). Similarly, Barakat et al. [1999] show that subjects are able to distinguish between Western vs. Eastern Arabic dialects significantly above chance based on intonation alone.

Hamdi et al. [2004] show that rhythmic differences exist between Western and Eastern Arabic. The analysis of these differences is done by comparing percentages of vocalic intervals (%V) and the standard deviation of intervocalic intervals (ΔC) across the two groups. These features have been shown to capture the complexity of the syllabic structure of a

adjacent frame; where a frame of voiced speech signal is represented as the summation of harmonics.

language/dialect in addition to the existence of vowel reduction. The complexity of syllabic structure of a language/dialect and the existence of vowel reduction in a language are good correlates with the rhythmic structure of the language/dialect; hence, such cues can be important for language/dialect identification [Ramus, 2002].

It has been shown, for language recognition, that the phonotactic and acoustic approaches are the most effective approaches. Thus these have received the most attention from researchers. This may be due to the fact that typical prosodic features capture, in addition to language/dialect dependent information, a great deal of speaker-dependent information (e.g., speaking rate and pitch range) and, more importantly, pragmatic and paralinguistic information, such as the emotional state of the speaker. Moreover, prosodic differences are typically suprasegmental phenomena (e.g., at the intonational and/or intermediate phrase level) which may require long test utterances during identification. Thus far, there is no obvious way to isolate only the prosodic information that can distinguish languages/dialects. In fact, we show in this thesis that global prosodic features plus prosodic features extracted at the pseudo-syllable level modeled in HMMs improve significantly over a purely phonotactic approach.

4.3 Materials – Four Broad Arabic Dialects

When training a system to recognize languages or dialects, it is essential to use training and testing corpora recorded under similar acoustic conditions. Otherwise, the trained models may capture channel specific information as opposed to linguistic differences. In this work, we test our approaches on the following four Arabic dialects.

- Iraqi Arabic, including three sub-dialects: Baghdadi, Northern, and Southern.
- Gulf Arabic, including three sub-dialects: Omani, UAE, and Saudi Arabic.
- Levantine Arabic, including four sub-dialects: Jordanian, Lebanese, Palestinian, and Syrian Arabic.
- Egyptian Arabic, including primarily Cairene Arabic.

We obtain corpora for the above dialects recorded in similar recording conditions from the Linguistic Data Consortium (LDC) CallHome and CallFriend corpora [Canavan and Zipperlen, 1996c; Canavan *et al.*, 1997b]. The data are spontaneous telephone conversations, produced by native speakers of the dialects, speaking with family members, friends, and unrelated individuals, sometimes about predetermined topics. Although some of the data have been annotated phonetically and/or orthographically by LDC, we do not make use of these annotations for our work.

In this work, we develop and compare multiple approaches for dialect recognition. Due to the nature of the differences between these approaches, we have used two slightly different corpora with different divisions of the data into training and test sets, denoted as Data I and Data II. Our initial approaches utilized Data I. In particular, during the development of an approach that models acoustic features directly, we found that the Levantine corpus appeared to have different recording conditions from the others. Therefore, for Data II, we decided to replace the Levantine corpus in further experiments with a different Levantine corpus. Moreover, in Data II, we decided to equalize the number of test speakers across multiple categories, including gender and landline vs. mobile phones, and to use multiple test trials from the same speaker.

4.3.1 Data I

We use the speech files of 965 speakers (about 41 hours of speech) from the Gulf Arabic conversational telephone Speech database for our Gulf Arabic data [Appen Pty Ltd, 2006a].[2] From these speakers we hold out 150 speakers for testing.[3] We use the Iraqi Arabic Conversational Telephone Speech database [Appen Pty Ltd, 2006b] for the Iraqi dialect, selecting 475 Iraqi Arabic speakers with a total duration of about 25.73 hours of speech. From these speakers we hold out 150 speakers[4] for testing. Our Levantine data consists of 1258 speakers

[2] We excluded very short speech files from the corpora.

[3] The 24 speakers in *devtest* folder and the last 63 files, after sorting by file name, in *train2c* folder (126 speakers). The sorting is done to make our experiments reproducible by other researchers.

[4] Similar to the Gulf corpus, the 24 speakers in *devtest* folder and the last 63 files (after sorting by filename) in *train2c* folder (126 speakers)

from the Arabic CTS Levantine Fisher Training Data Set 1-3 [Maamouri *et al.*, 2006]. This set contains about 78.8 hours of speech in total. We hold out 150 speakers for testing from Set 1.[5] For our Egyptian data, we use CallHome Egyptian and its Supplement [Canavan *et al.*, 1997b] and CallFriend Egyptian [Canavan and Zipperlen, 1996c]. We use 398 speakers from these corpora (75.7 hours of speech), holding out 150 speakers for testing.[6] (about 28.7 hours of speech.)

Unfortunately, as far as we can determine, there is no data with similar recording conditions for MSA. Therefore, we obtain our MSA training data from the TDT4 Arabic Broadcast News corpus. We use about 47.6 hours of speech. The acoustic signal was processed using forced-alignment with the transcript to remove non-speech data, such as music. For testing we again use 150 speakers, this time identified automatically from the GALE Year 2 Distillation evaluation corpus. Non-speech data (e.g., music) in the test corpus was removed manually. It should be noted that the data includes read speech by anchors and reporters as well as spontaneous speech spoken in interviews in studios and though the phone.

4.3.2 Data II

We use the speech of the 478 speakers from the Iraqi Arabic Conversational Telephone Speech corpus [Appen Pty Ltd, 2006b], holding out 20% of the speakers for testing. We use the 976 speakers from the Gulf Arabic Conversational Telephone Speech corpus [Appen Pty Ltd, 2006a], again holding out 20% of the speakers for testing. Our Levantine data consists of 985 speakers from the Levantine Arabic Conversational Telephone Speech corpus [Appen Pty Ltd, 2007], also holding out 20% of the speakers for testing. These three corpora were collected by the same company (Appen Pty Ltd) and appear to have been collected under similar conditions. Each of the corpora contains male and female speakers speaking by landline or mobile phones. Since it is likely that the distribution of these categories may influence the trained models, we decided to equalize the number of test speakers in each category. So, our test set for each of the three dialects include: 25% are selected randomly from the set of female speakers speaking on mobile phones; 25% selected from

[5] We use the last 75 files in Set 1, after sorting by name.

[6] The test speakers were from *evaltest* and *devtest* folders in CallHome and CallFriend.

male speakers speaking on mobile phones; 25% selected from females speaking on landline phones; and 25% selected from males speaking over landlines. For the Egyptian dialect corpus, we use the 280 speakers in CallHome Egyptian and its supplement [Canavan *et al.*, 1997b] for training. Attempting to test our system on different acoustic conditions, we employ a completely different corpus for testing: 120 speakers from CallFriend Egyptian [Canavan and Zipperlen, 1996c]. This corpus was collected under different conditions. The Egyptian data also includes male and female speakers, but it is not clear if the speakers used landlines, mobile phones, or both. All corpora are provided by the Linguistic Data Consortium (LDC).

To identify speech regions in the audio files, we segmented the files based on silence using Praat [Boersma and Weenink, 2001], using a silence threshold of -35db, with a minimum silence interval of 0.5s and minimum sounding intervals of 0.5s. All segments were used in training. In this work, we test our system on 30-second cuts of this corpus. Each cut consists of consecutive speech segments totaling 30s in length.[7] Multiple cuts are extracted from each speaker. For Iraqi, we have a total of 477 30s test cuts, and 801, 818, 1912 30s test cuts for Gulf, Levantine, and Egyptian, respectively.

4.4 Dialect Recognition Framework

Some of the approaches used previously for dialect recognition, particularly the Parallel-PRLM and GMM-UBM, have been employed with considerable success in the language recognition community. In this thesis, we test and analyze the performance of the Parallel-PRLM approach on dialect recognition in some detail in Chapter 5. In Chapter 6, we describe a new way to model the prosodic structure of dialects to improve the Parallel-PRLM approach. We further examine the effectiveness of the GMM-UBM approach, widely employed in the language/speaker recognition community, on the task of Arabic dialect recognition in Chapter 7. We also show how we can improve such an approach by applying a speaker adaptation technique. In Chapters 8 and 9, we introduce two novel approaches and show that these approaches, which are specifically designed to model subtle phonetic

[7] N.B. It is sometimes necessary to truncate speaker turns to achieve exactly 30 seconds.

dialectal differences, significantly outperform the others.

Hazen and Zue [1993] have designed an excellent formal probabilistic framework to incorporate different components which model the phonotactic, prosodic, and acoustic characteristics of languages for the task of language identification. We adopt this framework throughout this thesis to guide us in explaining the different approaches.

Let $D = \{D_1, D_2, ..., D_n\}$ represents the dialect set of interest of a predetermined language. Given a speech utterance U of a speaker, we denote the acoustic information of the utterance by two sequences: (1) $\vec{a} = \{\vec{a}_1, \vec{a}_2, ..., \vec{a}_T\}$, the frame-based vector sequence that encodes the wide-band spectral information of U. (2) $\vec{f} = \{\vec{f}_1, \vec{f}_2, ..., \vec{f}_T\}$, the frame-based prosodic feature vector sequence (i.e., F0 and intensity contours). If the task of interest is identification, then we would like to find the dialect D_i that can best matches U. Therefore, we can view the problem as a maximization process, as shown in (4.1).

$$\underset{i}{\operatorname{argmax}}\ P(D_i|\vec{a}, \vec{f}) \tag{4.1}$$

The expression in (4.1) can be viewed as the most general expression describing the dialect recognition task. Since every utterance contains an underlying sequence of linguistic units, Hazen and Zue attempted to incorporate these units into the framework. Let $C = \{c_1, c_2, ..., c_k\}$ represent the most likely linguistic unit sequence obtained from some system, and $S = \{s_1, s_2, ..., s_{k+1}\}$ represent the corresponding alignment segmentation (e.g., time offsets for each unit) in the utterance. If our linguistic units are phonemes, for example, then these units and segmentations can be obtained from the first best hypothesis of a phone/phoneme recognizer. Assuming that these sequences can be obtained independently of the dialect, it was shown that expression (4.1) can be reduced to (4.2), which is equivalent to the expression in (4.3), see [Hazen and Zue, 1993] for the detailed mathematical derivations.

$$\underset{i}{\operatorname{argmax}}\ P(D_i|C, S, \vec{a}, \vec{f}) \tag{4.2}$$

$$\underset{i}{\operatorname{argmax}}\ P(D_i)\ P(C|D_i)\ P(S, \vec{f}|C, D_i)\ P(\vec{a}|C, S, \vec{f}, D_i) \tag{4.3}$$

To simplify the modeling process, we can see that, instead of modeling one complicated expression in (4.2), we can model each expression in the four factors in (4.3), separately. For example, we note that the linguistic units and prosodic information are contained in separated terms. Throughout this thesis, our choice of linguistic units are phonemes (i.e., C is a sequence of phonemes); therefore, these four expressions are termed:

1. $P(D_i)$: The prior probability of the dialect.

2. $P(C|D_i)$: The phonotactic model.

3. $P(S, \vec{f}|C, D_i)$: The prosodic model.

4. $P(\vec{a}|C, S, \vec{f}, D_i)$: The acoustic model.

4.5 NIST Evaluation Framework

For our phonotactic and prosodic modeling approaches, we report dialect identification results using classification accuracy and an F-Measure for each class. However, in order to allow comparison of our results to those obtained by other recent dialect-recognition systems, we adopt the NIST language/dialect and speaker recognition evaluation framework for the rest of our evaluations. In this framework, we report detection results instead of identification. In the detection task, we are given a hypothesis and a set of test trials. We are asked to give a decision for each test trial to accept or reject the hypothesis, along with a confidence score. Employing these scores, we report our results using Detection Error Tradeoff (DET) figures, which plots false alarms versus miss probabilities, and Equal Error Rate (EER), the error rate when both false alarm and miss probabilities are equal [Martin *et al.*, 1997]. To plot an overall DET, our results are pooled across each pair of dialects with dialect prior equalized to discount the impact of different number of test trials in each dialect.[8]

[8] We use the NIST scoring software developed for LRE07: www.itl.nist.gov/iad/mig/tests/lre/2007

Chapter 5

Phonotactic Modeling (Approach I)

5.1 Introduction

The phonotactic approach to dialect recognition relies on the hypothesis that dialects differ in their phone sequence distributions. Arabic dialects differ in many respects, such as phonology, lexicon, and morphology; therefore, it is highly likely that they too differ in terms of phone sequence distribution and phonotactic constraints. Using the probabilistic framework in Chapter 4 and assuming that the prior distribution is uniform, the dialect recognition problem can simply be written as in (5.1) — i.e., in this approach, the prosodic and acoustic models are ignored.

$$\operatorname*{argmax}_{i} P(C|D_i) \tag{5.1}$$

5.2 PRLM Approach

A well-known method for modeling phonotactic constraints of languages is PRLM (Phone Recognition followed by Language Modeling) [Zissman, 1996]. In this approach, for dialect recognition, the phones of the training utterances of a dialect are first recognized using a single phone recognizer.[1] Then an N-gram model is trained on the resulting phone sequences for this dialect. This process results in an N-gram model (λ_i) for each dialect that models the dialect's distribution of phone sequence occurrences. During recognition, given a test speech segment, one runs the phone recognizer to obtain the phone sequence C for this segment and then computes the likelihood of the phone sequence given the N-gram dialect model. For example, if $N = 3$, then the likelihood is computed as shown in (5.2).

$$P(C = c_1, c_2, ..., c_k; \lambda_i) = P(c_1; \lambda_i)P(c_2|c_1; \lambda_i) \prod_{j=3}^{k} P(c_j|c_{j-1}, c_{j-2}; \lambda_i) \tag{5.2}$$

The dialect with the N-gram model that maximizes the likelihood is selected as the hypothesized dialect of the given speech utterance, as shown in the expression (5.1).

5.3 Parallel PRLM Approach

Parallel PRLM is an extension to the PRLM approach, in which multiple (m) parallel phone recognizers, each trained on a different language, are used instead of a single phone recognizer [Zissman, 1996]. For training, one runs all phone recognizers in parallel on the set of training utterances of each dialect. An N-gram model is trained on the outputs of each phone recognizer for each dialect. Thus if we have n dialects, m x n N-gram models are trained. During testing, given a test utterance, we run all phone recognizers on this utterance and then compute the likelihood of the output phone sequence of each phone recognizer given the corresponding N-gram model. Finally, the likelihoods are fed to a combiner to determine the hypothesized dialect. It should be noted that this approach assumes all streams of phones from the different phone recognizers to be independent. Therefore, expression (5.1) can be written as in (5.3), where C_j is the phone sequence obtained from phone recognizer j and λ_i^j is N-gram model j of dialect i.[2]

[1] The phone recognizer is typically trained on one of the languages/dialects being identified. However, a phone recognize trained on any language can be a good approximation, since languages/dialects may share many phones in their phonetic inventory.

[2] Note that likelihoods in (5.3) have to be normalized first.

$$\underset{i}{\text{argmax}}\ P(C_1;\lambda_i^1)\ P(C_2;\lambda_i^2)\ ...\ P(C_k;\lambda_i^m) \tag{5.3}$$

Instead of using such a simple combination and classification criteria (i.e., maximum of the product of normalized likelihoods), we can make use of a back-end classifier. In our work, we have experimented with multiple discriminative classifiers, such as logistic regression, SVM and neural networks. We have found that a logistic regression classifier is superior. The recognition system with a back-end classifier is illustrated in Figure 5.1. There are multiple advantages of utilizing a back-end classifier: (1) The likelihood scores may not be comparable across phone recognizers (2) some phone recognizers may be less effective than others on the task. A logistic regression classifier, for example, finds the optimal weights for linearly combining the normalized likelihoods to *discriminate* dialects.

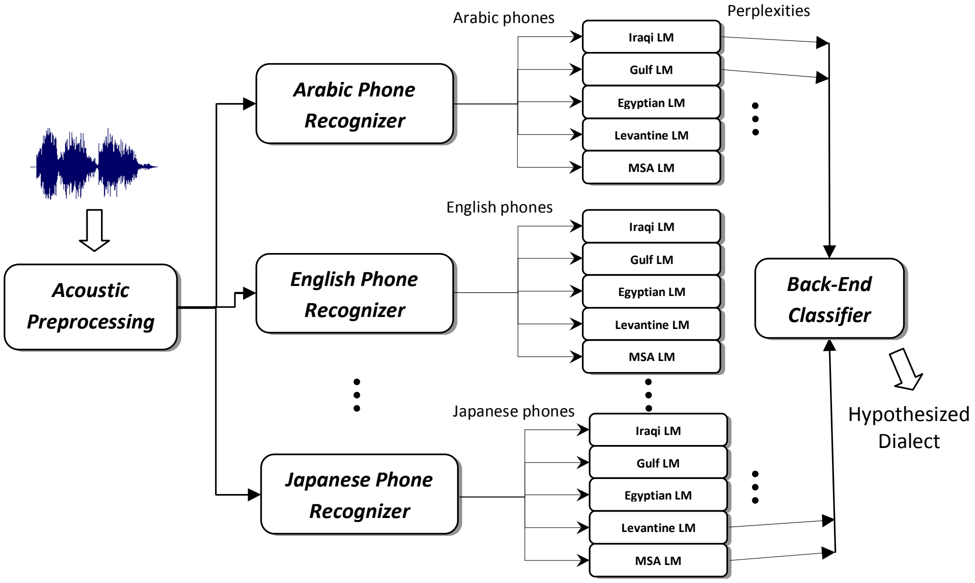


Figure 5.1: Parallel phone recognition followed by language modeling (Parallel PRLM) for dialect recognition.

The idea behind using multiple phone recognizers as opposed to only one is to allow the system to capture more phonetic differences that might be crucial for distinguishing dialects.

Particularly, since the phone recognizers are trained on different languages, they may be able to model different vocalic and consonantal systems, hence a different phonetic inventory. For example, an MSA phone recognizer typically does not distinguish the phonemes /g/ and /ʒ/; however, an English phone recognizer does. This phoneme is an important cue to distinguishing Egyptian Arabic from other Arabic dialects. Moreover, phone recognizers are prone to many errors; relying upon multiple phone streams rather than one may lead to a more robust model overall.

5.4 Phone Recognizers

In our experiments, we have used phone recognizers for English, German, Japanese, Hindi, Mandarin, and Spanish, from a toolkit developed by Brno University of Technology.[3] These phone recognizers are trained on the OGI multilanguage database [Muthusamy *et al.*, 1992] using a hybrid approach based on Neural Networks and Viterbi decoding without language models (open-loop) [Matejka *et al.*, 2005].

Since Arabic dialect recognition is our goal, we hypothesize that an Arabic phone recognizer will also be useful, particularly since other phone recognizers do not cover all Arabic consonants, such as pharyngeals and emphatic alveolars. We build three MSA phone recognizers: An open-loop phone recognizer which does not distinguish emphatic vowels from non-emphatic **(ArbO)**. This is the same as XPR, described in Chapter 3. Using the exact settings (training data and design) of XPR, we build two other phone recognizers: Open-loop with emphatic vowels **(ArbOE)**, and a phone recognizer with emphatic vowels and with a bi-gram phone language model **(ArbLME)**. We add a new pronunciation rule to the set of rules described in Chapter 3 to distinguish emphatic vowels (vowels in the context of emphatic consonant, see Section 2.3.1) from non-emphatic ones when generating the pronunciation dictionary. In total we employ 9 (Arabic and non-Arabic) phone recognizers.

[3] http://speech.fit.vutbr.cz/en/software/phoneme-recognizer-based-long-temporal-context

5.5 Evaluation of PRLM and Parallel PRLM

We conduct three experiments. The first is an attempt to classify four colloquial Arabic dialects using the Parallel PRLM approach. In the second experiment, we compare the performance of PRLM and Parallel PRLM approaches on the four dialects. Finally, in the third, we evaluate the parallel PRLM approach when including MSA as the fifth "dialect" – a five-way classification task. In all the experiments, we use the SRILM toolkit [Stolcke, 2002] to train backoff trigram models with Witten-Bell smoothing. The 150 test speakers, described in Section 4.3 (Data I), of each dialect are first decoded using the phone recognizer(s) to phone sequences. Then the perplexities of the corresponding trigram models on these sequences are computed and given to the logistic regression classifier.[4] Instead of splitting our held-out data into test and training sets to train/test the back-end classifier, we report our results with cross validation. To analyze how dependent our system is on the duration of the test utterance, we report the system accuracy and the F-measure of each class for different durations (5s – 2m). The longer the utterance, the better we expect the system to perform.

5.5.1 Four Arabic Dialect Recognition (Parallel PRLM)

In our first experiment, we test our system on four colloquial Arabic dialects (Gulf, Iraqi, Levantine, and Egyptian) in Data I. As mentioned above, we use the phone recognizers to decode the training data to train the nine trigram models per dialect (9x4=36 trigram models). We report our 10-fold cross validation results on the test data in Figure 5.2. We can observe from these results that, regardless of the test-utterance duration, the best distinguished dialect among the four dialects is Egyptian (F-Measure of 90.1% with 30s test utterances), followed by Levantine (F-Measure of 79.9% with 30s). The most confusable dialects, according to the classification confusion matrix, are those of the Gulf and Iraqi Arabic (F-Measure of 65.9%, 65.7%, respectively with 30s). This confusion is consistent with dialect classifications that consider Iraqi a sub-dialect of Gulf Arabic. Using this framework

[4] In this work, we have not employed feature selection, which make our results slightly different from our published work [Biadsy *et al.*, 2009].

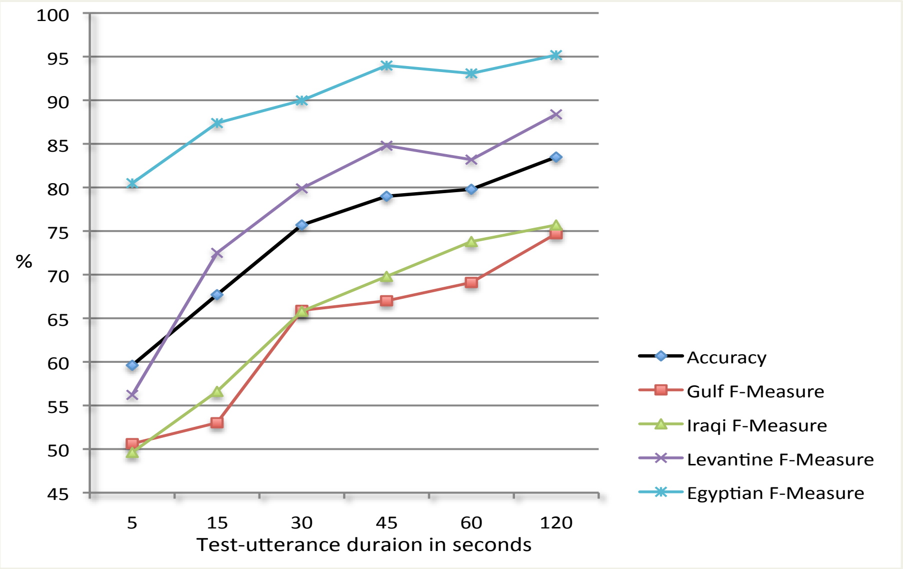


Figure 5.2: The accuracies and F-Measures of the four-way classification task with different test-utterance durations

on 2-minute utterances, we obtain a classification accuracy of 83.5%. Note that if instead of using a back-end classifier, we compute the expression in (5.3), we obtain substantially lower accuracy of 65.5% (for 2-minute utterances).

5.5.2 PRLM vs. Parallel PRLM for Dialect Recognition

In this experiment, we compare the PRLM approach (one stream of phones) versus the Parallel PRLM approach (using 9 streams of phones) for the task of four Arabic dialects. For the PRLM, we use only the phone recognizer with emphatic vowels (ArbOE), and the nine phone recognizers as described in previous section for the Parallel PRLM. In this experiment, we report results with 25-fold cross validation to allow more reliable statistical significance testing. The results are shown in Figure 5.3. We observe that Parallel PRLM outperforms the PRLM approach in all test duration conditions with statistical significance in almost all cases except for the 120 second utterances. These results are consistent with previous results in the language recognition literature.

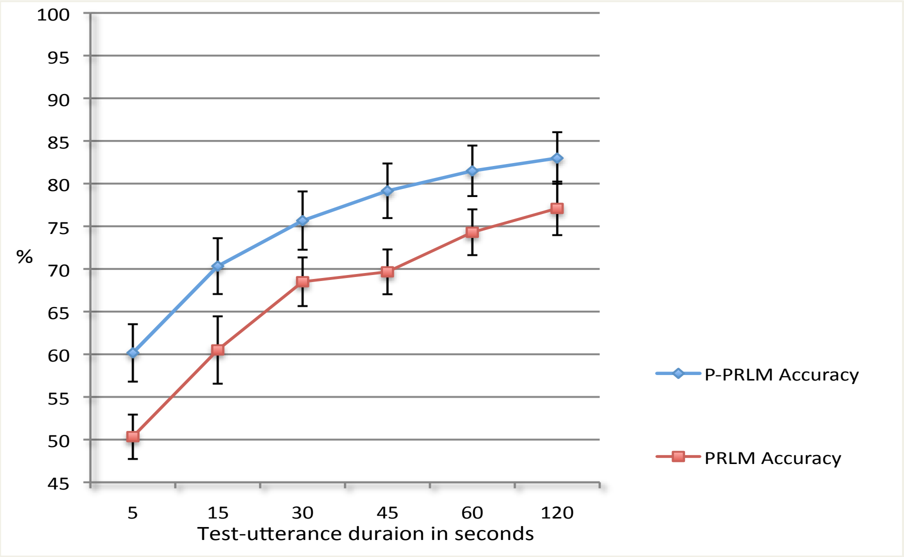


Figure 5.3: The PRLM versus the Parallel PRLM approach for the four-way classification task with different test-utterance durations. The bars represent the standard error with 0.05 significance level

5.5.3 Dialect Recognition with MSA

Considering MSA as a dialectal variant of Arabic, we are also interested in analyzing the performance of our system when including it in our classification task. In this experiment, we add MSA as a fifth class. We perform the same steps described above for training, using the MSA corpus described in Section 4.3 (Data I). For testing, we use also our 150 hypothesized MSA speakers as our test set. Interestingly, in this five-way classification, we observe that the F-Measure for the MSA class in the cross-validation task is always above 98% regardless of the test-utterance duration, except for the 15s case (94.6%), as shown in Figure 5.4.

It would seem that MSA is rarely confused with any of the colloquial dialects: it appears to have a distinct phonotactic distribution. This explanation is supported by linguists, who note that MSA differs from Arabic dialects in terms of its phonology, lexicon, syntax and morphology, which appears to lead to a profound impact on its phonotactic distribution [Holes, 2004]. Similar to the four-way classification task, Egyptian was the most easily

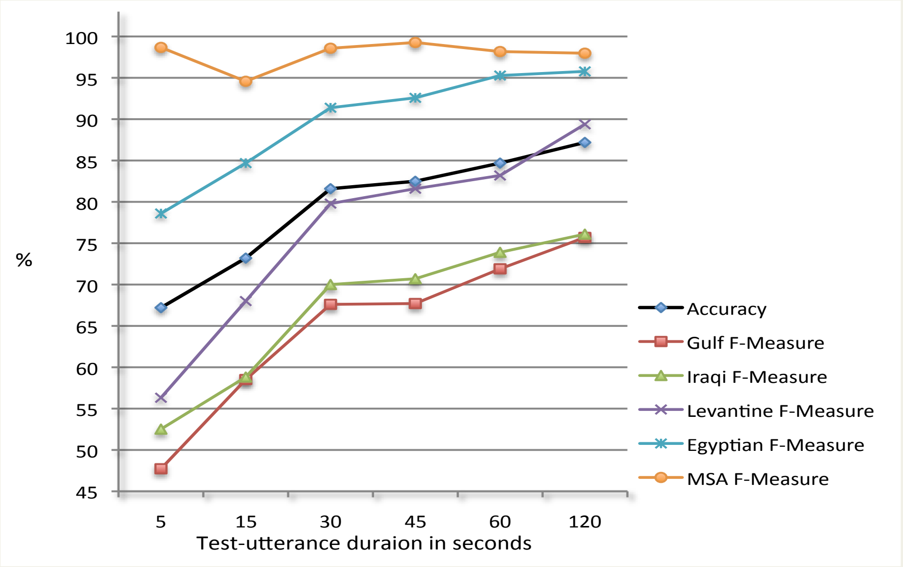


Figure 5.4: The Accuracies and F-Measures of the five-way classification task with different test-utterance durations

distinguished dialect (F-Measure=91.4%, with 30s test utterance) followed by Levantine (79.8%), and then Iraqi and Gulf (67.6% and 70%, respectively). Due to the high MSA F-Measure, the five-way classifier can also be used as a binary classifier to distinguish MSA from colloquial Arabic (Gulf, Iraqi, Levantine, and Egyptian) reliably.

However, it should be noted that our classification results for MSA might be inflated for several reasons: (1) The MSA test data were collected from Broadcast News, which includes read (anchor and reporter) speech, as well as telephone speech (for interviews). (2) The identities of the test speakers in the MSA corpus were determined automatically, and so might not be as accurate since we do cross-validation. As a result of the high recognition rate of MSA, the overall accuracy in the five-way classification task is higher than that of the four-way classification.

5.6 Conclusions

In this chapter, we have analyzed the performance of well-known language recognition phonotactic-based approaches (PRLM and Parallel-PRLM) on distinguishing our four broad

Arabic colloquial dialects plus MSA. We have found that these dialects significantly differ in terms of their phonotactic distributions. Parallel-PRM can identify Arabic dialects with considerable accuracy, especially when employing a back-end classifier. Importantly, we have observed that Parallel-PRLM rarely confuses MSA with any other colloquial dialects, suggesting that MSA has its own distinct phonotactic constraints. Consistent with previous results in the language recognition literature, we have seen that Parallel-PRLM significantly outperforms the PRLM approach in most test-durations on Arabic dialects as well.

Chapter 6

Prosodic Modeling (Approach II)

6.1 Introduction

The prosodic modeling approach for dialect recognition relies on the hypothesis that dialects differ in their prosodic structure. Using the probabilistic framework described in Chapter 4 and assuming that the prior distribution is uniform, the dialect recognition problem can simply be written as in (6.1) — i.e., in this approach, the phonotactic and acoustic models are ignored. In addition to suggesting a method to model the prosodic structure of dialects, we also focus our attention in this section on identifying and analyzing prosodic differences across four Arabic dialects. In particular, we attempt to answer the following questions: (1) Do dialects differ in terms of their prosodic structure?; if so, (2) what are the individual prosodic cues that make dialects different?; (3) How can we model the prosodic structure of a dialect?; (4) How well does a dialect recognition system that relies only on prosodic features perform?; and finally, (5) how much can prosody contribute to the recognition task when combined with a phonotactic approach?

$$\underset{i}{\operatorname{argmax}}\; P(S, \vec{f}|C, D_i) \tag{6.1}$$

6.2 Prosodic Differences Across Dialects

In this section, we identify *global prosodic features* that differ significantly across our four Arabic dialects. We randomly select 398 speakers from each dialect corpus from (Data I) and examine the first 2 minutes of speech from each speaker. We first segment the speech files based on silence. We assume that each non-silent segment is a valid *speech segment*; inspection of a random sample of the output of this process shows this assumption to be reasonable. In this work, several of our prosodic features are calculated at the syllable level; therefore, we next syllabify the speech segments. Since, to our knowledge, there are no automatic syllabification systems for Arabic dialects that employ only acoustic information, we employ a pseudo-syllabification approach which has been employed in previous work [Rouas, 2007; Timoshenko and Hoge, 2007]. We define a pseudo-syllable as a cluster of optional consonants followed by a single vowel (i.e., C*V). To identify vowels and consonants, we run our open-loop phone recognizer (ArbO), described in Section 5.4, and map all six MSA vowels to V and all other phones to C. Note that we have time boundaries of the syllables from our phone recognizer.

6.2.1 I. Pitch Features Across Dialects

To test whether dialects differ in their pitch variation, we compute the mean pitch range for each speaker by first Z-normalizing the entire F0 contour and then computing the average of the F0 maxima in all the speaker's segments.[1] Using the normalized F0 contour, we also compute the pitch register across dialects; this is computed as the average of the difference between the F0 maximum and F0 minimum over all the speech segments of the speaker. Similarly, we extract the average of the F0 minimum of all speech segments of the speaker. We also compute the standard deviation of the entire (unnormalized) F0 contours of the speaker to test if one dialect employs more dynamic intonational contours than other dialects.

Previous work has suggested that H peaks may align earlier in Egyptian formal Arabic

[1] We use the mean of F0 maxima for pitch range instead of the absolute maximum, to reduce the sensitivity to errors introduced by the pitch tracking algorithm.

(within the stressed syllable) than in Egyptian colloquial Arabic [Hellmuth and El Zarka, 2007]. To test whether Arabic dialects differ in the alignment of the pitch peaks to syllables, we compute the mean distance (in seconds) of pitch maxima from the beginning of the syllable, normalized by the duration of the syllable. This value is between 0 and 1. (Currently, we do not attempt to distinguish stressed syllables from unstressed.) We then compare these prosodic features for each pair of dialects, using Welch's t tests. Table 6.1 shows the differences we have observed in the data for each pitch feature across all 6 pairs of dialects. X* indicates that dialect X has a greater mean for that feature than does the other dialect with significance level of 0.05; **, with significance level of 0.01; and ***, with 0.001.

Dialect 1	Dialect 2	Pitch Register	Pitch Range	Pitch Min	Pitch SDev	Pitch Peak Alignment
Gulf	Iraqi	I***	I***	G**	–	I***
Gulf	Levantine	L***	L***	–	G .07	G**
Gulf	Egyptian	E***	–	G***	G***	E***
Iraqi	Levantine	–	L .067	L*	I***	I***
Iraqi	Egyptian	I***	I***	I .056	I***	–
Levantine	Egyptian	L***	L***	L***	L***	E***

Table 6.1: Comparing global pitch features between dialect pairs

We see from these results that Levantine and Iraqi speakers tend to speak with higher pitch range and more expanded pitch register than Egyptian and Gulf speakers. In addition, Gulf speakers tend to use a more compressed pitch register than Egyptian speakers. Moreover, Iraqi and Gulf intonation show more variation than Egyptian and Levantine. Nonetheless, the intonational contours of Levantine speakers vary significantly more than that of Egyptian speakers. Pitch peaks within pseudo-syllables in Egyptian and Iraqi are shifted significantly later than the pitch peaks in Gulf and Levantine. However, Levantine speakers tend to shift their pitch peaks earlier in syllables than do Gulf speakers.

6.2.2 II. Durational and Rhythmic Features Across Dialects

We analyze dialects' timing features using Ramues' rhythmic measures [Ramus, 2002]. Particularly, we compare percentages of vocalic intervals (%V), the standard deviation of rvocalic intervals (ΔV), and the standard deviation of intervocalic intervals (ΔC) across pairs of dialects. These measures have been shown to capture the complexity of the syllabic structure of a language/dialect in addition to the existence of vowel reduction. Languages/dialects that have a high variability of consonantal intervals are likely to have more clusters of consonants, which lead to more complex syllables. The complexity of syllabic structure of a language/dialect and the existence of vowel reduction in a language/dialect are good correlates with the rhythmic structure of the language/dialect. In this work, we identify vocalic intervals using our phone recognizer. A sequence of consecutive vowels are considered as a single vocalic interval. Similarly a sequence of consecutive consonants is considered as one intervocalic interval. Again, we use Welch's t test to indicate significant differences in features between each dialect pair. Table 6.2 again shows our results.

Dialect 1	Dialect 2	ΔC	ΔV	%V	Speaking Rate
Gulf	Iraqi	–	G***	G***	G***
Gulf	Levantine	G*	–	–	G**
Gulf	Egyptian	G***	E**	E***	E**
Iraqi	Levantine	I***	L***	L***	–
Iraqi	Egyptian	I***	E***	E***	E***
Levantine	Egyptian	L***	E .1	E***	E***

Table 6.2: Comparing global durational features between dialect pairs

We observe that both Gulf and Iraqi have significantly higher variation in their intervocalic intervals than Levantine and Egyptian. Assuming that our automatically obtained pseudo-syllables are good approximations of true syllables, we may conclude that Gulf and Iraqi dialects tend to have more complex syllabic structure. Also, Egyptian has the lowest variation in its intervocalic intervals, suggesting that it has the least complex syllabic structure.

Egyptian tends to have longer and more variation in vocalic intervals than other dialects, which may account for vowel reduction and quantity contrasts. These features suggest that some of these dialects do in fact differ in their rhythmic structure, empirical confirmation of previous phonological hypotheses.

We also want to test the effect of speaking rate on distinguishing our Arabic dialects. Speaking rate is computed here as the number of pseudo-syllables per second. We see that Egyptian speakers are the fastest speakers followed by Gulf speakers. Iraqi and Levantine are the slowest speakers, with comparable rates.

6.3 Modeling Prosodic Patterns

Although we have found major differences between dialects in prosodic and rhythmic variation, we suspect that the global features described above are not sufficient to capture aspects of the prosodic structure of a dialect. These features do not, for example, capture specific contextual, segmental and sequential patterns, such as the shape of intonational contours and the distribution of different contour types in a dialect. We believe that modeling sequences of local prosodic features using sequential models, such as HMMs, may be more effective in modeling the prosodic patters of a dialect. To model sequential prosodic structure, we extract *five* different sequences from each speech segment in our training data: mean F0 ($\vec{f}_{mean}$), pitch slope ($\vec{f}_{slope}$), pitch peak alignment ($\vec{f}_{peaks}$), RMS (Root Mean Square) intensity ($\vec{e}$), and duration ($\vec{d}$). Each sequence consists of two-dimensional feature vectors. Each vector is extracted from prosodic data within pseudo-syllables. These features are illustrated in Figure 6.1 and described below.

To test whether dialects differ in the characteristics of their intonational contours, we extract three types of sequences from the Z-Normalized F0 contour. We calculate the mean of the F0 values within each pseudo-syllable and compute the *deltas* of these means to approximate the first derivative of the F0 contour (this feature is denoted as **I** in Figure 6.1); we define delta here as the difference between each two consecutive values. To model pitch slope, we fit a linear regression given the values of the Z-normalized F0 contour in each pseudo syllable, and extract the angle of the regression line (denoted as **II** in the figure).

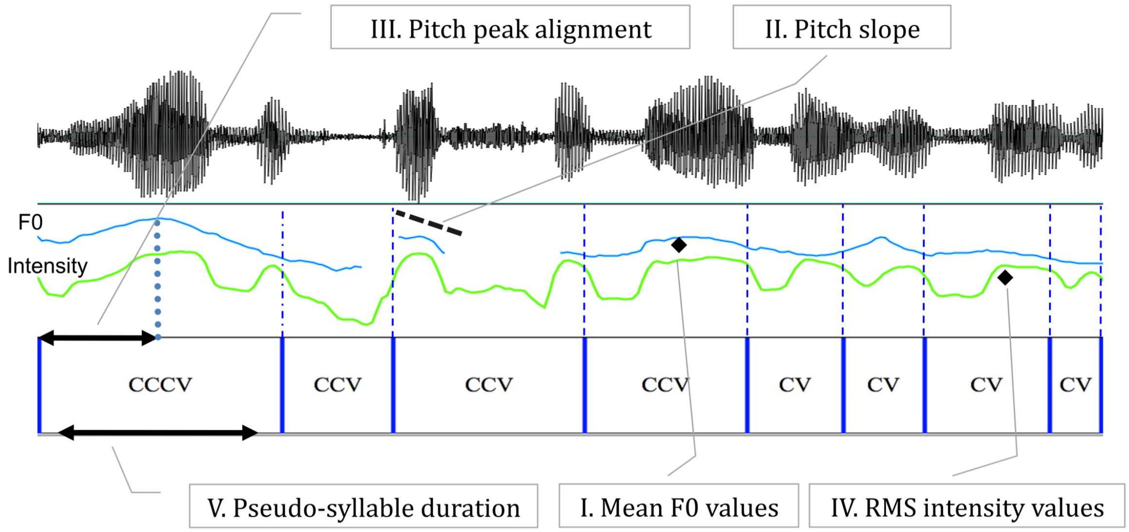


Figure 6.1: Local prosodic features extracted from the pseodo-syllables in a speech segment

We also add the deltas of these angles. For pitch peak alignment, we extract the location in time (starting from the onset of the syllable) of the F0 peak within pseudo syllables (denoted as **III**). The values of these features are between 0 and 1. We also compute the delta of these locations.

Intensity features play an important role in prosodic events [Rosenberg and Hirschberg, 2006]. Therefore, for each speech segment, we first Z-normalize the intensity contour and then extract the RMS of the intensity values within pseudo-syllables (denoted as **IV** in the figure). We also add the deltas of these RMS intensity features.

As mentioned in Section 4.2, Arabic dialects have been shown to differ in their rhythmic structure. We approximate the rhythm of a dialect by modeling the sequence of the log of the duration of each pseudo-syllable (denoted as **V**). Similar to the other sequences, the delta of these log durations is included in the feature vector. This modeling of rhythm is somewhat similar to [Timoshenko and Hoge, 2007], but that work models rhythm using a joint multinomial distribution of two consecutive durations instead of an HMM of log durations and deltas, described next.

Now, we turn our attention to modeling the prosodic structure of each dialect using the prosodic features described above for the task of dialect recognition. To model the probability distribution in (6.1), as mentioned above, we extract multiple prosodic sequences at the level of pseudo-syllables. First, note that the expression in (6.1) is equivalent to the expression in (6.2). Since we assume here that the phonotactic model is uniform, we obtain the expression in (6.3). Now, instead of modeling this complicated probability distribution, we extract only useful prosodic feature sequences, by employing some parameterization function of $\{\vec{f}$, S, and $C\}$.[2] Assuming that our prosodic features are limited to $\{\vec{f}_{mean}, \vec{f}_{slope}, \vec{f}_{peaks}, \vec{e}, \vec{d}\}$, the expression in (6.3) is equivalent to the expression in (6.4). In this work, we assume that these sequences are statistically independent, thus expression (6.4) can be written as in (6.5).[3]

$$\underset{i}{\operatorname{argmax}}\ P(\vec{f}, S, C|D_i)/P(C|D_i) \tag{6.2}$$

$$\underset{i}{\operatorname{argmax}}\ P(\vec{f}, S, C|D_i) \tag{6.3}$$

$$\underset{i}{\operatorname{argmax}}\ P(\vec{f}_{mean}, \vec{f}_{slope}, \vec{f}_{peaks}, \vec{e}, \vec{d}|D_i) \tag{6.4}$$

$$\underset{i}{\operatorname{argmax}}\ P(\vec{f}_{mean}|D_i)P(\vec{f}_{slope}|D_i)P(\vec{f}_{peaks}|D_i)P(\vec{e}|D_i)P(\vec{d}|D_i) \tag{6.5}$$

An obvious approach that can be used to model each of these sequences is an ergodic HMM, described in the next section. Although HMM assumes conditional independence between the sequence elements, it has proven to be robust in similar scenarios. Note that

[2] Note that this function uses S and C to pseudo-syllabify the utterance and then extract prosodic features for each pseudo-syllable.

[3] In fact, we have built a model without this assumption, but our approach which involves a back-end classifier (described below) performs significantly better than that of one joint model.

Hazen and Zue [1993] extract only the F0 contour and assume that they are independent of the phone durations and identities. They model the F0 contour using a multinomial distribution in which they assume that the F0 points are statistically independent.

6.4 HMM Settings

For each dialect, we model each of the five sequence types mentioned above using an ergodic continuous HMM with GMM observation distribution for all states with diagonal covariance matrices for all Gaussian components.[4] The state transition matrix (A) and initial state distributions (π) in all HMMs are initialized uniformly, and the Gaussian mixture components of all the states are initialized by running k-mean clustering first. The number of states and number of Gaussians are determined empirically. For all the F0 HMMs (I–III), we use four hidden states with one Gaussian per state. For the intensity HMMs (IV), we use six states and two Gaussian components per state, and for the durational HMMs (V), we use 3 states and one Gaussian per state. We have an HMM for each pair of dialect and sequence type. Since we analyze four dialects and five sequence types, we have 20 HMMs in total. All HMMs are trained using the Baum-Welch algorithm on the training data in Section 4.3 (Data I). We use the HMM Matlab toolkit [Murphy, 2004] for training and decoding.

6.5 Evaluation Using Prosodic Features

In this section, we describe a system for classifying the four Arabic dialects using the global and sequential prosodic features described above, which we then compare to the parallel PRLM system described in the previous section. Finally, we combine these two systems to see if prosodic features provide information that phonotactics does not.

We first evaluate the effectiveness of the global features described in Section 6.2 for dialect recognition. We use the 150 test speakers in Data I from each dialect to train a logistic classifier that uses only the nine global features. Four-way 10-fold cross-validation

[4] We have experimented with full covariance matrices instead, but generally we have not observed improvements over diagonal matrices.

classification shows that, with these features only, we obtain an accuracy of 54.83%. F-Measures of the classes are shown in Table 6.3; the chance baseline is 25%.

Feature Type	Acc (%)	Gulf (F_1)	Iraqi (F_1)	Lev (F_1)	Egy (F_1)
Chance baseline	25.0	-	-	-	-
Nine global prosodic features	54.8	41.2	53.6	56.5	65.3
+ Vowel duration mean & SDev.	60.0	52.7	57.1	62.8	66.9
+ Sequential prosodic modeling	72.0	68.9	66.4	72.9	79.2
Phonotactic classifier (Parallel PRLM only)	83.5	74.7	75.7	88.4	95.2
Phonotactic & prosodic features (one classifier)	81.5	74.1	74.6	86.3	90.2
Combining phonotactic & prosodic classifiers	86.3	79.5	81.5	89.5	94.9

Table 6.3: Four-way 10-fold cross-validation dialect recognition results for our 600 speakers, with different feature sets; F_1 is the F-Measure. Test utterance duraion is 2 minutes.

We have also observed that different dialects lengthen certain vowels more than others, so we include the mean and standard deviation of the durations of each vowel type from a speaker as features in our classifier. When we analyze the errors of our phone recognizer, we also observe that glottal stops and vowels are often confused, so we include the duration and standard deviation of glottal stop durations as well. Thus, we have fourteen additional features: the mean and standard deviation of 6 vowels and the glottal stop phone. When we add these duration features we obtain a significant increase in accuracy 60%. All F-measures also show some increase, as shown in Table 6.3. It should be noted that the vowel duration features do not perform well alone; the accuracy of the dialect recognition system using the fourteen features alone is only 44.16%.

To test the usefulness of our sequential prosodic features on dialect recognition, we extract the feature-vector sequences of each sequence type from each dialect and train an HMM on the training corpus for each of our dialects. In total, we have 20 (4 dialects x 5 sequence types) HMMs. Given a speaker's utterance, we first extract each sequence type and compute the likelihood of this sequence given each of the five corresponding HMMs. Again using the held-out 150 speakers for each dialect, if we calculate the expression in (6.5), we achieve an accuracy of only 38.0% for our four-way classification task.

Similar to the Parallel-PRLM back-end classifier, we hypothesize that finding optimal weights for combining the likelihoods to discriminate dialects is also important. Therefore,

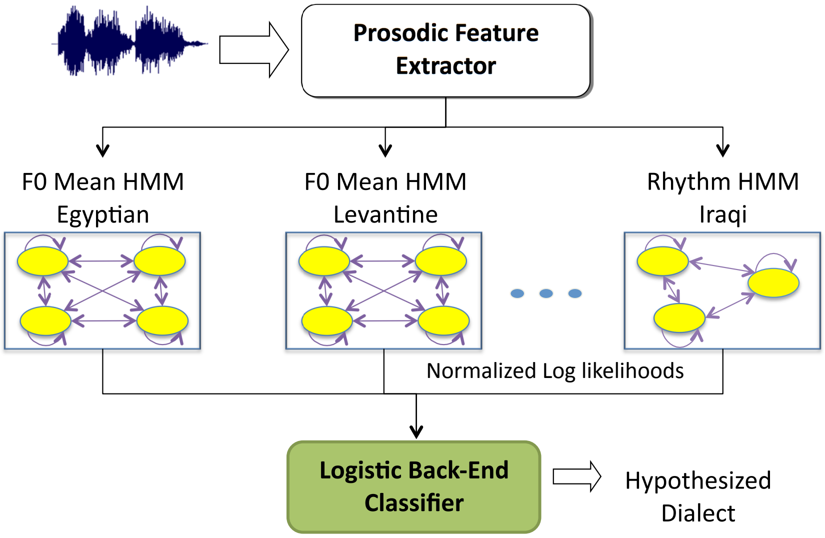

Prosodic Feature Extractor
F0 Mean HMM Egyptian
F0 Mean HMM Levantine
Rhythm HMM Iraqi
Normalized Log likelihoods
Logistic Back-End Classifier
Hypothesized Dialect

Figure 6.2: Dialect recognition for local prosodic features

we make use of a logistic regression back-end classifier where the normalized log-likelihoods of each utterance are the features (4 dialects x 5 HMMs = 20 features).[5] We report 10-fold cross-validation results over the 600 speakers held out from HMM training. The recognition framework is illustrated in Figure 6.2. Using a back-end classifier, we obtain an accuracy of 63.8%, a substantial improvement. Another advantage of using a back-end classifier as opposed to the product of the likelihoods is that it allows us to include additional features. In fact, when we add the global prosodic features, we obtain a significant increase in accuracy of 72% (Table 6.3).

6.6 Combining the Phonotactics and Prosodic Features

We have seen that prosodic features when used alone are valuable features for identifying Arabic dialects. We also have observed, in Chapter 5, that phonotactic features so far are superior at distinguishing dialects. Now we examine whether prosodic features add new information that may improve dialect classification. If so, how can we best combine

[5] The log-likelihoods are normalized by the length of the corresponding sequence.

phonotactic and prosodic information?

Recall that we have two back-end logistic classifiers, one for the phonotactic approach (see Section 5.3) and another for the prosodic approach. If, instead of training the two separately, we train a single classifier that includes both phonotactic and prosodic information, we obtain an accuracy of 81.5% – somewhat lower than the accuracy of the phonotactic classifier alone. We speculate that the reason for this lower performance may be a data sparsity issue, since we increase the feature dimensionality but still perform 10-fold cross-validation on only 600 instances.

So, instead of training one classifier that combines all features, we combine the posterior probabilities of the two classifiers by multiplying the posterior probabilities and then returning the class with the maximum score. We found that this approach outperforms the sum and max combination strategies [Kittler *et al.*, 1998]. Using this approach, we obtain a significant (p-value=.022) increase in accuracy (86.33%) over the phonotactic approach alone (Table 6.3). We have also obtained similarly significant increase in accuracy when using 15, 25 and 50 -fold cross-validation.

It should be noted that the percentage of instances that are *in*correctly classified by the phonotactic classifier but *correctly* classified by the prosodic classifier is 9.5%. Thus, the upper bound accuracy that could be obtained by using the phonotactic and the prosodic classifiers together would be 93% (9.5 + 83.5). Further research is required to find a better method for combining phonotactic and prosodic features. Similar to the phonotactic approach, with the combined systems, we also observe that the most distinguishable dialect among our four dialects is Egyptian, followed by Levantine, and still the most confusable dialect pairs are Iraqi and Gulf Arabic.

6.7 Distinguishing Dialects using Phonotactic versus Prosodic Features

In this section, we attempt to test the following hypothesis:

> *The easier it is to distinguish a pair of Arabic dialects using phonotactic features, the easier it is to distinguish them using prosodic features.*

We test this hypothesis by analyzing the correlation between the performance of our phonotactic-based classifier and our prosodic-based classifier on all pairs of dialects. In particular, we train two binary logistic regression classifiers for each pair of dialects. The first classifier makes use only of phonotactic features, and the second makes use only of prosodic features (Global + HMM log-likelihoods). We compute the accuracies of 10-fold cross-validation of the two classifiers; thus we have a 2D point for each pair of dialects.[6] We plot these 2D points for the six pairs shown in Figure 6.3. We find that there is a significant and strong correlation between these accuracies (p-value=0.01, r^2=0.82, ρ=0.91).

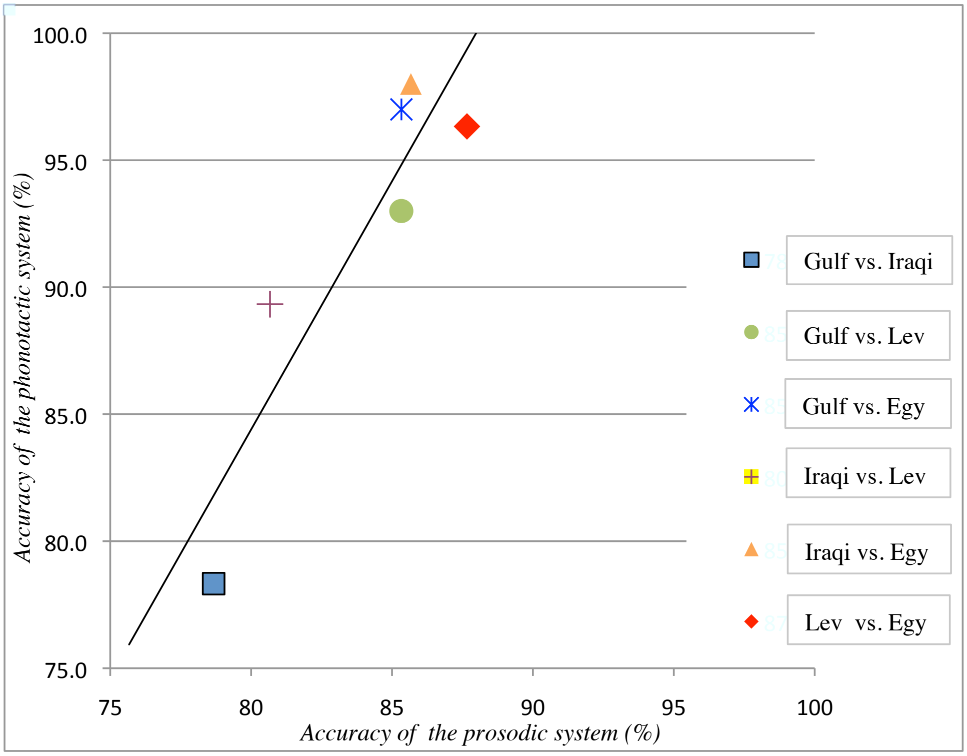


Figure 6.3: Linear Regression between the accuracy of the phonotactic approach and the prosodic approach for each pair of dialects

[6] We use 2-minute utterances for each system.

It is known that there are certain correlations between these two domains of phonetic/phonotactic and prosodic information. Vowels are distinguished in part by differences in their intrinsic F0, for example. This relationship between phonotactic and prosodic structures may be seen in the relationship that exists between syllabic structure, which is partly captured by phonotactic constraints, and rhythmic structure. In our own work, we have found that $\%V$ and $\Delta\ V$ in Egyptian Arabic are higher than they are in all other Arabic dialects. Also, we have seen that Egyptian is the most easily distinguished dialect using either a phonotactic or a prosodic approach. We hypothesize that Egyptians greater percentage of vocalic intervals may allow opportunities for a greater range of pitch patterns. Nonetheless, the correlation we observe between the performance of our two classifiers is still quite striking. It is possible that this correlation reflects an important underlying relationship between the prosodic and phonotactic structures of a dialect and they may in fact constrain each other. More careful studies however will be needed to validate this hypothesis.

6.8 Conclusions

We have shown empirically that four Arabic dialects (Gulf, Iraqi, Levantine, and Egyptian) exhibit significant differences from one another in terms of characteristics of their prosodic structure, including pitch range, register, and pitch dynamics, as well as differences in their rhythmic structure, speaking rate, and vowel durations. We have demonstrated that we can utilize these prosodic features to automatically identify the dialect of a speaker with considerable accuracy. Modeling sequences of local prosodic features at the level of pseudo syllables using HMMs significantly improves accuracy when combined with global prosodic features, resulting in an accuracy of 72.0%. Such accuracy strongly indicates that prosody alone carries significant information for distinguishing dialects. Note that this information is also available to human listeners attempting to identify dialects, and suggests that these subjects can rely on prosody to do that. This has been corroborated by perceptual studies in two German dialects as well as Eastern vs. Western Arabic dialects, as discussed in Section 4.2.

Our prosodic modeling approach can also significantly improve a system that utilizes phonotactic features only, resulting in an accuracy from 83.5% to 86.3% . Although our analyses and modeling techniques here are specifically done for Arabic dialects, our methodology is general enough to be applied to other dialects of other languages. We have also observed that the more difficult it is to distinguish a pair of dialects using the phonotactic approach, the more difficult it is to distinguish using only prosodic features.

Chapter 7

Acoustic Modeling (Approach III)

7.1 Introduction

Acoustic modeling has received a great deal of attention in the past decade for both language and speaker recognition systems, due to its simplicity and relatively good performance. The acoustic modeling approach for dialect recognition relies on the hypothesis that dialects differ in terms of their spectral distribution. According to phonologists, Arabic dialects have been shown to differ in the vowel and consonantal spaces as well as in some subtle phonetic realizations (see Chapter 2). Therefore, it is also highly likely that the spectral distributions will significantly differ across Arabic dialects. Again, using the probabilistic framework in Chapter 4 and assuming that the prior distribution is uniform, the dialect recognition problem can be written as in (7.1) — i.e., the phonotactic and prosodic models are ignored.

$$\underset{i}{\operatorname{argmax}}\ P(\vec{a} = a_1, a_2, ..., a_T | C, S, \vec{f}, D_i) \tag{7.1}$$

7.2 Acoustic Feature Extraction

The acoustic features we extract for this approach are the same features employed in IBM's Attila Arabic ASR system [Soltau *et al.*, 2009]. The front end is a 13-dimensional PLP with cepstral mean and variance normalization (CMVN). Each frame is spliced together with four preceding and four succeeding frames and then LDA is performed to yield 40-dimensional feature vectors. We use the LDA matrix derived for IBM's Attila Arabic ASR system here [Soltau *et al.*, 2009]. Hence, a_i (where $1 \leq i \leq T$) in (7.2) is a 40D PLP vector.

7.3 GMM Approach

Most acoustic-based language/dialect recognition systems employ, at some point, a Gaussian Mixture Model (GMM) to model the acoustic space of each language/dialect. GMM is a well-studied statistical model. GMMs are computationally inexpensive, and, typically, the standard acoustic features used tend to be locally normally distributed and uncorrelated. In this approach, each language/dialect's acoustic frames are modeled using a separate GMM. What makes this approach simple is the assumption that the acoustic-frame sequence $\vec{a}$ is statistically independent of the linguistic units C, segmentation S, and prosodic features $\vec{f}$. Hence, the expression in (7.1) can be reduced to (7.2). Furthermore, the acoustic frames are assumed to be i.i.d. (independent and identically-distributed). Thus we obtain (7.3). Under the assumption that the acoustic distribution is a GMM with m mixtures, we get the expression in (7.4), where ω_{ij}, μ_{ij}, and Σ_{ij} are the weight, mean, and covariance matrix of Gaussian j (where $1{\leq}j{\leq}m$) of dialect i respectively, and $\mathcal{N}$ represents the pdf (probability density function) of the normal distribution.

$$\underset{i}{\operatorname{argmax}}\ P(\vec{a} = a_1, a_2, ..., a_T | D_i) \tag{7.2}$$

$$\underset{i}{\operatorname{argmax}}\ \prod_{t=1}^{T} P(a_t | D_i) \tag{7.3}$$

$$\underset{i}{\operatorname{argmax}}\ \prod_{t=1}^{T} \sum_{j=1}^{m} \omega_{ij} \mathcal{N}(a_t; \mu_{ij}, \Sigma_{ij}) \tag{7.4}$$

7.4 GMM-UBM

The parameters of the GMM of each dialect could be estimated using the well-known Expectation Maximization (EM) algorithm. However, instead, a "large" GMM-Universal Background Model (GMM-UBM) is first trained to represent the dialect-independent ("universal") distribution of acoustic data. Afterwords, a separate GMM for each dialect is derived by MAP (Maximum A-Posteriori) adapting the trained GMM-UBM to the acoustic training data of that dialect (see next section for MAP adaptation). The advantages of using such adaptation over running the EM algorithm to train each dialect model (GMM) separately are as follows: We obtain a tighter coupling between the dialect model and the UBM. This coupling has shown to outperform the decoupled models for speaker recognition [Reynolds *et al.*, 2000]. Moreover, all dialect models have the same initialization parameters, which are the same as the UBM's. In addition, MAP adaptation combines the robustly estimated UBM parameters with the dialect model parameters. This leads to more robust estimates of dialect models for those dialects with insufficient training data. Finally, training a new dialect model is faster than running the EM again on each dialect – it requires only a few adaptation iterations.

7.4.1 MAP Adaptation for GMMs

For the past decade, adapting the UBM parameters using MAP (or so-called Bayesian) adaptation has become a standard technique in the speaker and language recognition communities. We next describe the formulas to adapt the GMM parameters.[1]

Let $X = \{x_1, ..., x_T\}$ be the training data of dialect k to which we are adapting the UBM. MAP adaptation is like the EM algorithm in consisting of two steps. The first step is identical to the expectation step, in which the sufficient statistics of X are computed for each mixture component i in the UBM, see (7.5)–(7.8).[2]

[1] The formulas and material presented in this subsection are based on [Reynolds *et al.*, 2000]. For more details about the mathematical derivations of these formulas, see [Gauvain and Lee, 1994].

[2] We assume diagonal matrices, where σ_i^2 is the variance vector of Gaussian component i. We denote $\mathbf{x^2}$ as $diag(xx^T)$.

$$P(i|x_t) = \frac{\omega_i \mathcal{N}(x_t; \mu_i, \sigma_i^2)}{\sum_{j=1}^{m} \omega_j \mathcal{N}(x_t; \mu_j, \sigma_j^2)} \tag{7.5}$$

$$n_i = \sum_{t=1}^{T} P(i|x_t) \tag{7.6}$$

$$E_i(\mathbf{x}) = \frac{\sum_{t=1}^{T} P(i|x_t) x_t}{n_i} \tag{7.7}$$

$$E_i(\mathbf{x^2}) = \frac{\sum_{t=1}^{T} P(i|x_t) x_t^2}{n_i} \tag{7.8}$$

The second step in adaptation is unlike the one in EM. In adaptation, the new sufficient statistics are then linearly combined with the old sufficient statistics using a data-dependent mixing factors ($\{\alpha_i^w, \alpha_i^m, \alpha_i^v\}$) to obtain the adapted parameters $\hat{\omega_i}, \hat{\mu_i}, \hat{\sigma_i}^2$ (for each Gaussian i), see (7.9)–(7.11). γ is computed over all adapted mixture weights to ensure they sum to one.

$$\hat{\omega_i} = [\alpha_i^w n_i / T + (1 - \alpha_i^w)\omega_i]\gamma \tag{7.9}$$

$$\hat{\mu_i} = \alpha_i^m E_i(\mathbf{x}) + (1 - \alpha_i^m)\mu_i \tag{7.10}$$

$$\hat{\sigma_i}^2 = \alpha_i^v E_i(\mathbf{x^2}) + (1 - \alpha_i^v)(\sigma_i^2 + \mu_i^2) - \hat{\mu_i}^2 \tag{7.11}$$

Note that there is a data-dependent adaptation coefficient α_i^ρ, where $\rho \in \{w, m, v\}$, for each mixture and each parameter in the above equations. This coefficient is defined in (7.12), where r^ρ is relevance factor for parameter ρ.

$$\alpha_i^\rho = \frac{n_i}{n_i + r^\rho} \tag{7.12}$$

7.5 Scoring

We describe in this section how we can obtain confidence scores to be used in plotting the DET curves (see Section 4.5). First, we denote the feature vector extracted for a given test trial r, as $\mathcal{O}_r$. For example, in the GMM-UBM approach, $\mathcal{O}_r$ is the acoustic-frame sequence $\vec{a}$ of trial r, in (7.2). Every test trial is given a confidence score for belonging to target dialect D_t. Assuming that the dialect priors are equal, the posterior probability of $\mathcal{O}_r$ can be reduced to the expression in (7.13). We use these posterior probabilities to represent the detection scores used in plotting DET curves, similar to [Matejka *et al.*, 2006]; where $\mathcal{D}$ is the set of dialects of interest, $p(\mathcal{O}_r|\lambda_{D_x})$ represents the likelihood of $\mathcal{O}_r$ given the model λ_{D_x} of dialect D_x, and τ_r normalizes duration differences across trials.

In the standard NIST Language Recognition Evaluation (LRE), the task is a pairwise language/dialect detection task; therefore, for score computation we can make use of the knowledge that an utterance belongs to either the target or non-target dialect. So, in our first scoring scheme, PairScoring, we normalize by the sum of the likelihoods under the target and non-target dialect models only — i.e., $\mathcal{D}$ in (7.13) contains only D_t and D_{nt}, the non-target dialect. However, in the second scoring scheme, AllScoring, we do not use this knowledge. Instead, we normalize by the sum of the likelihoods of $\mathcal{O}_r$ under every model to represent the final score — i.e., $\mathcal{D}$ in (7.13) contains all our dialects.

$$P(D_t|\mathcal{O}_r) = \frac{p(\mathcal{O}_r|\lambda_{D_t})^{\tau_r}}{\sum_{D_x \in \mathcal{D}} p(\mathcal{O}_r|\lambda_{D_x})^{\tau_r}} \tag{7.13}$$

7.6 Evaluation of GMM-UBM

Employing the evaluation framework described in Section 4.5, we evaluate the standard GMM-UBM approach using the acoustic features described in Section 7.2. We first extract these acoustic features to obtain the 40-dimensional PLP vectors for the training and test sets in Data II. We use an equal number of training frames from three dialects (Iraqi, Gulf, and Levantine) to ML-train the UBM with 2048 Gaussian components (with diagonal covariance matrices), using the EM algorithm. Note that it has previously been shown that 2048 components achieve the best performance for language and speaker recognition tasks.

A GMM (λ_{D_x}) is created for each dialect (D_x) by MAP adapting only the means of the UBM using the entire training data for that dialect. We run the MAP adaptation in 5 iterations with a relevance factor of $r^m = r = 16$.[3] These settings are similar to [Torres-Carrasquillo *et al.*, 2008]. In this work, we do not employ fast scoring [Reynolds *et al.*, 2000; Wong *et al.*, 2000].

During testing, we calculate the scores as in (7.13), where $\mathcal{O}_r$ represents the sequence of 40D PLP features of trial r, and $p(\mathcal{O}_r|\lambda_{D_x})$ represents the likelihood of $\mathcal{O}_r$ given GMM λ_{D_x} of dialect D_x, and τ_r is the inverse of the number of frames in the sequence $\mathcal{O}_r$. We use the test data of the four dialects, described in Section 4.3 (Data II), to test the performance of the GMM-UBM approach on 30s cuts. We test the two scoring schemes described in Section 7.5. Using AllScoring, which uses all the scores from all GMM models, we obtain an EER of 20%. We get a significant improvement when utilizing PairScoring: an EER of 15.3%. The overall DET curve using PairScoring is shown in Figure 7.1.

7.7 Context-Dependent Phone Recognizer

We next show how we can improve the GMM-UBM approach by employing feature space Maximum Likelihood Linear Regression (fMLLR) adaptation. For such an adaptation, we need to provide the fMLLR algorithm either word or phone hypothesis for each utterance. Employing phone hypotheses, we build our own phone recognizer which uses the front-end described in Section 7.2. We build a continuous HMM-based triphone context-dependent (CD) phone recognizer using IBM's Attila system [Soltau *et al.*, 2009]. This phone recognizer is trained on MSA using 50 hours of GALE speech data of broadcast news and broadcast conversations, downsampled to 8Khz. Our phone recognizer consists of 230 context-dependent acoustic models and a total of 20,000 Gaussians. We use one acoustic model for silence, one for non-vocal noise and another to model vocal noise. Therefore, in total, we have 227 CD-phones. The set of CD-phones is automatically generated by using a decision tree which asks questions about left and right contexts of each triphone. Contexts

[3] r controls the balance between old and new estimates. The value $r = 16$ is employed in most recent language/dialect and speaker recognition works.

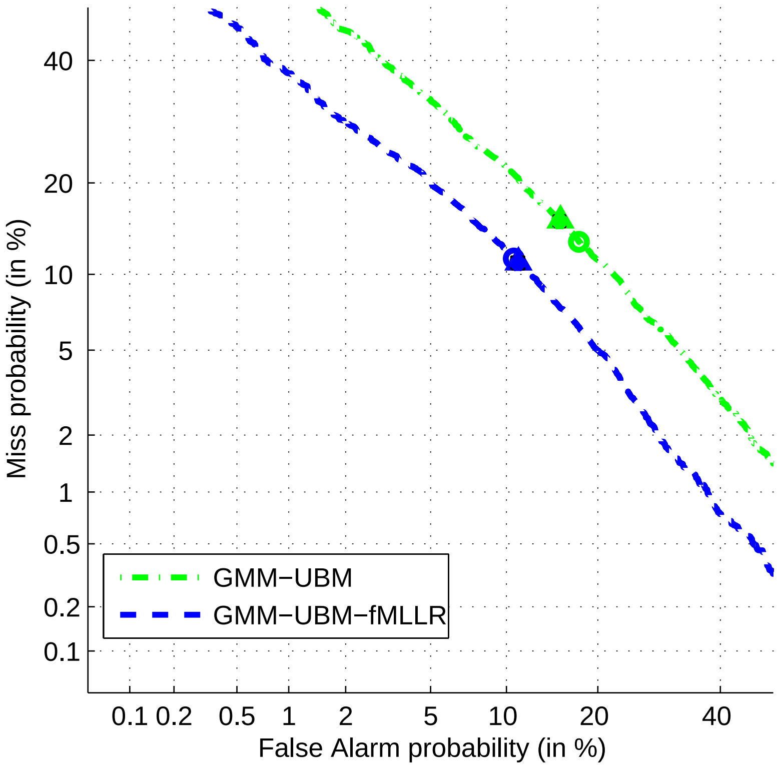


Figure 7.1: Overall DET curves for GMM-UBM vs. GMM-UBM-fMLLR

with the smallest acoustic difference are clustered together.

All CD-phone HMMs consist of 3 states, except for the the MSA short vowels (/a/ /i/ /u/) which consist of only 2 states.[4] All state observation densities are GMMs. We utilize a unigram language model of phones trained on MSA. We do not use higher order of n-gram to avoid bias for any particular dialect. The pronunciation dictionary and MSA phonetic inventory used in this work are generated as described in Chapter 3.

The phone-recognizer is a two-pass system. In the first pass, we obtain the most likely phone sequence hypothesis. The second pass uses this hypothesis to perform model adaptation, followed by decoding. In this work, we first apply fMLLR followed by MLLR adaptation, given the most likely phone sequence hypothesis. The resulting CD-phones are

[4] It has been previously shown that 2 states for short vowels as opposed to 3 significantly improves ASR word error rate [Soltau *et al.*, 2009].

exemplified by the CD-phone /r/: [Voiced-Consonant & !Glide]-**/r/**-[Front Vowel].

7.8 GMM-UBM with fMLLR Adaptation

It has been shown that the GMM-UBM approach can be improved by applying some normalization/transformation techniques for the acoustic signal. For example, Wong and Sridharan [Wong and Sridharan, 2002] and Torres-Carrasquillo et al. [2008] have shown that Vocal Tract Length Normalization (VTLN), to remove speaker-dependent features, improves language and dialect recognition results. In addition, channel compensation techniques to retain only language dependent information have been shown to significantly improve performance (c.f. [Campbell *et al.*, 2006b; Torres-Carrasquillo *et al.*, 2008]).

It has been demonstrated that fMLLR adaptation method helps remove channel effects for ASR [Rennie and Dognin, 2008]. In this work, we apply fMLLR to transform the feature vectors given the phone hypotheses. Specifically, we first run the CD-phone recognizer described above to obtain the most likely phone sequences. Afterwords, we use the phone sequences to first estimate the fMLLR affine transformation $[\mathbf{A}\ \vec{b}]$ for each speaker. We then use this matrix to transform the acoustic data of the speaker ($\hat{a}_i = [\mathbf{A}\ \vec{b}][a_i^T\ \mathbf{1}]^T$). Finally, we use the transformed frames ($\hat{a}_i$) as new features in the GMM-UBM approach. To the best of our knowledge, fMLLR has not been employed for the task of language/dialect recognition in such framework. We term this approach GMM-UBM-fMLLR.

Applying the same settings of the GMM-UBM experiment in Section 7.6, but with fMLLR adaptation, we achieve an EER of 15.8% with the AllScoring scheme. Similar to GMM-UBM without adaptation, we obtain significantly better results when utilizing PairScoring: an EER of 11.0%. The GMM-UBM approach with fMLLR, interestingly, provides us with significantly higher results when compared to GMM-UBM without adaptation. The comparison of DET curves between GMM-UBM with and without fMLLR adaptation is shown in Figure 7.1; both use PairScoring. We speculate that this substantial improvement could be due to the reduction of channel effects, which may have resulted in more compact GMMs that focus on linguistic information as opposed to channel variations. We also hypothesize that the fMLLR matrices of speakers may be "more similar" within the same dialect than those of speakers across dialects, and a result can lead to better separation of dialects.[5]

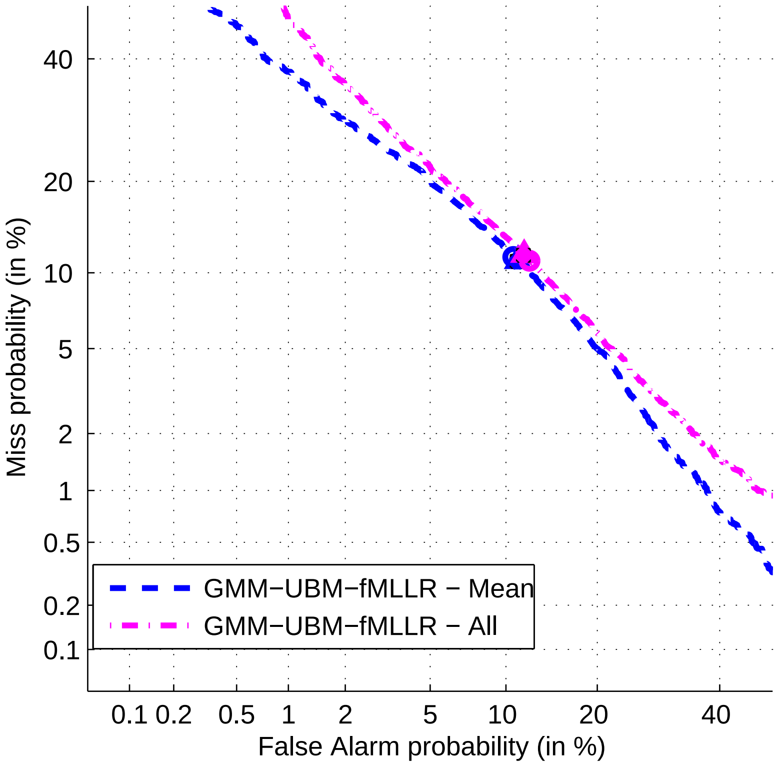


Figure 7.2: Overall DET curves for the GMM-UBM-fMLLR with adapting means only vs. GMM-UBM-fMLLR with adapting all parameters: means, covariances, and weights.

Throughout this work, we MAP adapt only the means of the Gaussians in GMMs. It has been shown that, in the GMM-UBM framework, adapting, in addition, the Gaussian covariance matrices and weights makes no significant improvement over adapting the means only [Reynolds *et al.*, 2000]. We validate these results using our fMLLR adapted features for the task of Arabic dialect recognition as well. We MAP adapt all the GMM-UBM parameters (means, weights and covariances), using a relevance factor $r^{\rho} = 16$, for each $\rho \in \{w, m, v\}$. We find, in fact, that such adaptation leads to higher error (EER: 11.7%)

[5] As future work, we will analyze this hypothesis by, for example, comparing the transformation matrix determinants of speakers across dialects to those of speakers within the same dialect.

compared to 11.0%, when adapting the means only. Nonetheless, the difference between the EERs is not statistically significant (see Figure 7.2).

7.9 Conclusions

We have seen in this chapter that the standard GMM-UBM approach, a well-known speaker and language recognition acoustic modeling approach, also performs well on our four Arabic dialects. This suggests that these dialects significantly differ in terms of their spectral distributions. We have improved this approach by applying a speaker adaptation technique to transform the feature space using fMLLR before employing the GMM-UBM approach. This feature transform substantially and significantly improves results (from EER of 15.3% to 11.0%). Moreover, consistent with the literature of speaker recognition, we have found that MAP adapting only the Gaussian means of the GMM-UBM as opposed to adapting all of its parameters yields comparable results.

Chapter 8

Discriminative Phonotactics (Approach IV)

8.1 Introduction

Thus far, we have seen three approaches that make use of phonotactic, prosodic, and acoustic features respectively. None of these approaches explicitly focuses on subtle context-dependent (CD) phonetic realization differences that may specifically contribute to distinguishing the dialects of interest. For example, the /r/ in Scottish English is trilled in some phonetic contexts but produced as an approximant in dialects such as American English. Considering such subtle differences is essential in particular when the inputs to the dialect recognition system are short utterances, since we may not be able to reliably observe higher level features, such as prosodic and/or phonotactic patterns. In this section, we introduce a new approach to dialect recognition that first classifies CD-phones to one of our dialects. The output of these classifiers is then used to *augment* the phonotactic features, which are subsequently given to a discriminative classifier to obtain dialect detection scores. We call this approach *Discriminative Phonotactics.* Note that, for this approach, we do not follow Hazen and Zue's [1993] probabilistic framework described above, since our models are based on discriminative classifiers as opposed to generative models.

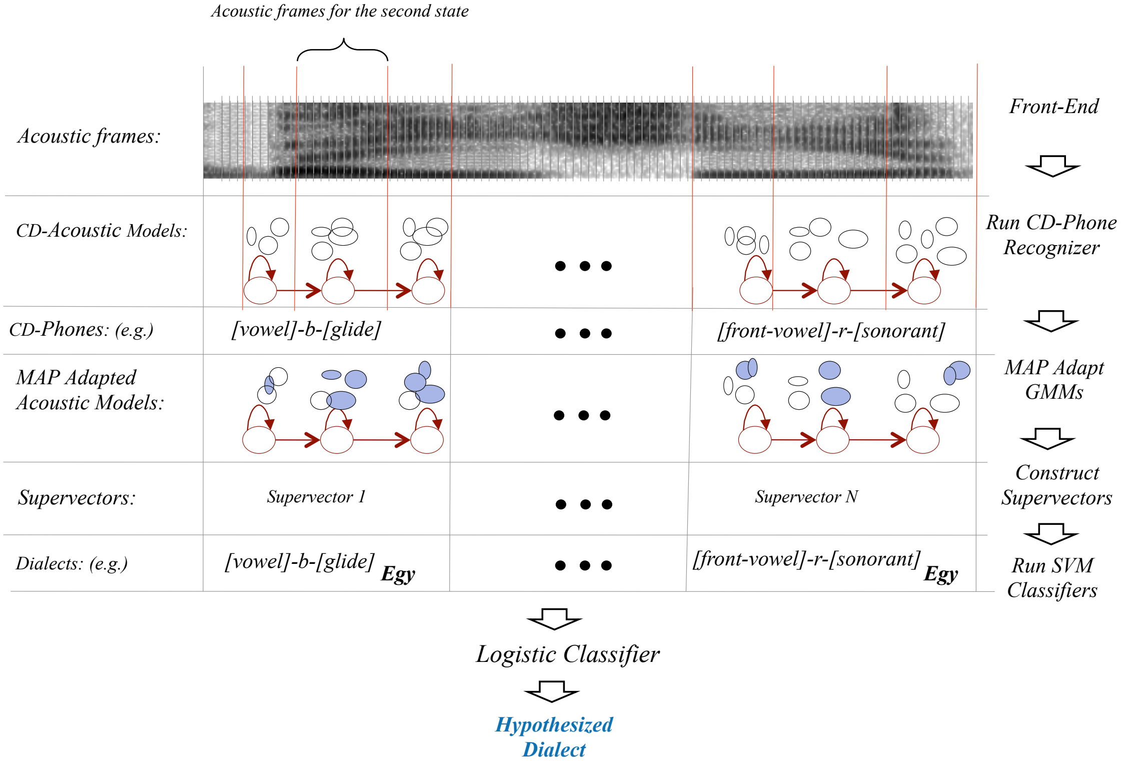


Figure 8.1: Discriminative Phonotactic Procedure

8.2 Context-Dependent Phone Classifiers

As noted above, dialects typically differ in some number of phonetic realizations in context. In this section, we describe an approach that allows us to classify each CD-phone instance in an utterance as belonging to one of our dialects. This approach is similar in spirit to the GMM-SVM approach introduced by Campbell et al. [2006a] for speaker verification. However, in our approach, we target the acoustic differences at the level of CD-phones as opposed to the differences in the overall acoustic data of a speaker, independent of linguistic units.

8.2.1 CD-Phone Representation

As illustrated in Figure 8.1, the first step in our approach, after front end processing, is to obtain the CD-phone sequence of a given speech utterance U. To do this, we run the CD-phone recognizer described in Section 7.7 to obtain the most likely phone sequence hypothesis. In the second step, for each CD-phone in the sequence, we extract the acoustic features aligned to each HMM state in the corresponding acoustic model. In other words, for each CD-phone instance in the phone sequence, we have one sequence of acoustic frames aligned to the first state in the HMM, and another frame sequence aligned to the second state. If the HMM has three states, then we have also another frame sequence aligned to the third state. Note that these features are extracted after normalization (CMVN) and fMLLR transformation. See the second row in Figure 8.1.

Recall that there is a GMM for each HMM state. For each CD-phone instance in the utterance, we adapt the GMMs of each of its HMM states. To do this, we use the acoustic frame alignments to the HMM states to MAP adapt each GMM in each state to the corresponding frames. That is, if the HMM has three states, then we get three new *adapted* GMMs for each CD-phone *instance* in the utterance; see the fourth row in Figure 8.1. In our implementation, we only adapt the means of the Gaussians using a relevance factor of $r = 0.1$. In the context of the GMM-UBM approach, the HMM can be viewed as the universal background model (UBM) of the CD-phone type.

To be able to classify a CD-phone instance as belonging to one of our dialects, we adopt the GMM-Supervector representation [Campbell *et al.*, 2006a] — but at the level of phone instances as opposed to a single vector for the entire utterance and HMMs instead of GMMs. We represent each CD-phone instance in the utterance by a *supervector* which is the result of stacking all the mean vectors of the two or all three adapted GMMs of the CD-phone HMM. The intuition is that the modified means of the adapted CD-phone GMMs 'summarize' the variable number of frames in a particular phone instance with a fixed-size representation.[1] It is also important to note that the supervector representation retains *some* of the phonetic structure of the CD-phone instance (albeit without the *complete* frame order). As observed in Chapter 6, the duration of vowels and certain consonants significantly differ across Arabic dialects. Therefore, we also include the phone duration as an additional feature in the supervector of each CD-phone.[2] These steps are summarized below.

[1] Supervectors can be with different lengths across CD-phone types.

[2] One could add additional prosodic features to the phone vector (similar to those in Chapter 6). We hypothesis that such features would be particularly useful for tonal dialects.

1. Run the CD-phone recognizer on utterance $U \Rightarrow$ CD-phone sequence
2. For each CD-phone instance:
   (a) Extract the acoustic features aligned to each HMM (of the corresponding CD-phone type) state
   (b) MAP adapt each GMM of each HMM state using the aligned frames ⇒ Adapted GMMs
   (c) Stack all the Gaussian mean vectors of the adapted GMMs and the phone duration in one vector ⇒ **Supervector**

8.2.2 CD-Phone Classification

Now, we make use of the above CD-phone representation to build a discriminative classifier at the CD-phone level. For training, we apply the procedure described above on the training data to obtain a set of supervectors for each CD-phone type from each dialect. Using these sets of supervectores, we train a binary discriminative classifier for each CD-phone type for each pair of dialects. From our 227 CD-phones, we thus have a total of 227 binary classifiers for each pair of dialects. In our implementation, we train SVM classifiers with RBF kernel.[3] We have found that an SVM with such a kernel performs significantly better than an SVM with a linear kernel and also better than a logistic regression classifier for the vast majority of the 227 classifiers. During testing, given a CD-phone instance with its frame alignment, we apply the procedure described above to extract its supervector, and then run the corresponding SVM classifier to classify this CD-phone into one of our dialects.

8.3 Automatic Extraction of Linguistic Knowledge

There are several uses of our CD-phone classification framework. First, we can utilize it to automatically extract linguistic knowledge, specifically the phonetic cues that may distinguish one of our dialects from another. We are particularly interested in knowing which phones in which contexts are realized differently across dialects. An empirical measure

[3] In our implementation, we use the LibSVM and LibLinear toolkits [Chang and Lin, 2001].

of the classification performance of each CD-phone classifier provides us with a measure of how the realization of a CD-phone is distinguishable across pairs of dialects.[4]

To extract these phonetic cues, we conducted the following experiment. We split the training speaker set (in Data II) of each dialect into halves. We use the first half to train the CD-phone classifiers for each pair of dialects and the second to test each classifier's performance. We randomly balance the number of test instances so that a chance baseline is 50%. Using the test instances, we apply the binomial test procedure to identify those CD-phone classifiers that perform on the test set with a significance level of 0.05. We report on this performance in Table 8.1 where we show the weighted accuracy of the classifiers that perform significantly better than chance for each dialect pair. We observe that the Egyptian dialect has the highest number of top performing classifiers under our definition.

Dialect Pair	Num. of * classifiers	Weighted accuracy (%)
Egyptian/Iraqi	195	70.9
Egyptian/Gulf	196	69.1
Egyptian/Levantine	199	68.6
Levantine/Iraqi	172	63.96
Gulf/Iraqi	166	61.77
Levantine/Gulf	179	61.53

Table 8.1: Number of CD-phone classifiers out of the 227 that performed significantly higher than chance for each pair of dialects (* significance level of 0.05)

We report the accuracy of the CD-phone classification results in Table 8.2-8.4 for the 10 most and 3 least accurate classifiers for some of our dialect pairs (with significance level of 0.05).[5] The third column in these tables contains the number of instances used in the classification task per dialect. The top 10 CD-phones can be viewed as those that best

[4] Note that other methods (such as Kullback-Leibler divergence) can be used to quantify differences between adapted dialect acoustic models. However our approach uses held out data instead of "distance" between models. Also our accuracy measures can be more easily interpreted.

[5] See Chapter 3 for the MSA phonetic symbols used in this work.

distinguish between a pair of dialects. We found, for example, that some consonants in the context of central vowels can be useful cues to distinguish dialects. Moreover, the phoneme /*k*/ is one of the top 10 cues for distinguishing between Iraqi and Levantine. This might be due to the consistent replacement of the MSA /*k*/ sound to /*ch*/ by the Iraqi dialect.[6] These empirical findings can be useful for dialectologists as well as speech scientists and engineers.

CD-Phone ([l-context]–phone–[r-context]	Accuracy	#
[∗]–***sh***–[∗]	71.1	6302
[SIL]–***a***–[∗]	70.3	3935
[SIL]–***?***–[Central Vowel]	68.7	1323
[∗]–***j***–[∗]	68.5	3722
[! Central Vowel]–***s***–[! High Vowel]	68.5	1975
[Nasal]–***A***–[Anterior]	68.1	5459
[!SIL & ! Central Vowel]–***E***–[!Central Vowel]	67.8	3687
[Central Vowel]–***m***–[Central Vowel]	66.7	2639
[!Voiced Cons. & !Glottal & !Pharyngeal & !Nasal & !Trill & !w & !Emphatic]–***A***–[Anterior]	66.4	11857
[∗]–***k***–[Central Vowel]	66.4	1433
...	...	...
[!SIL & !Central Vowel]–***G***–[!Central Vowel]	57.5	852
[!A]–***h***–[Back Vowel]	57.0	409
[!Vowel & !SIL]–***m***–[!Central Vowel & !Back Vowel]	56.2	300

Table 8.2: The 10 most and 3 least accurate CD-phone classifiers for Levantine/Iraqi dialects (with significance level of 0.05)

It should be noted that substantially more accurate phonetic cues can be obtained by making use of orthographic transcripts in the system instead of using a phone recognizer. In other words, we can do forced-alignment to obtain the phone sequences and then train/analyze the CD-phone classifiers from that. However, we currently lack such orthographic transcripts and/or a pronunciation dictionary that maps our colloquial dialect transcripts onto a shared phonetic inventory.

[6] Note that /*ch*/ and /*k*/ are modeled as one phoneme in the phone recognizer.

CD-Phone ([l-context]–phone–[r-context]	Accuracy	#
[!Central Vowel & !Unvoiced Cons.]–***t***–[SIL]	71.2	473
[∗]–***sh***–[∗]	67.9	6302
[SIL]–***w***–[Central Vowel]	67.3	745
[!Central Vowel]–***H***–[Central Vowel]	67.0	1234
[SIL]–***a***–[∗]	66.5	3935
[!Central Vowel]–***s***–[!Hight Vowel]	66.2	1975
[SIL]–***b***–[!Central Vowel & !Front Vowel]	66.1	505
[!Central Vowel & !SIL]–***b***–[Central Vowel]	66.1	750
[!SIL & !Central Vowel]–***E***–[Central Vowel]	65.8	1480
[!SIL & !Central Vowel]–***E***–[! Central Vowel]	65.7	3687
...	...	...
[Strident]–***u***–[∗]	55.7	380
[Glottal Stop]–***a***–[∗]	55.3	515
[Pharyngeal]–***A***–[!SIL & !Anterior]	55.1	484

Table 8.3: The 10 most and 3 least accurate CD-phone classifiers for Gulf/Iraqi dialects

CD-Phone ([l-context]–phone–[r-context]	Accuracy	#
[∗]–***sh***–[∗]	80.2	8127
[Central Vowel]–***H***–[Central Vowel]	77.4	1980
[SIL]–***f***–[!Front Vowel]	76.5	612
[SIL]–***m***–[Central Vowel]	75.8	2547
[∗]–***T***–[Central Vowel Vowel]	75.5	1145
[!Central Vowel]–***s***–[!High Vowel]	75.3	3396
[SIL]–***a***–[*]	75.1	7411
[h]–***A***–[Anterior]	74.5	1370
[!Central Vowel & !Unvoiced Cons.]–***t***–[SIL]	74.4	857
[SIL]–***w***–[Central Vowel]	74.1	1534
...	...	...
[Front Vowel]–***h***–[!Back & !Central Vowels]	59.0	183
[Central Vowel]–***?***–[Central Vowel]	58.4	353
[!Vowel & !SIL]–***m***–[SIL]	57.5	389

Table 8.4: The 10 most and 3 least accurate CD-phone classifiers for Egyptian/Gulf dialects

8.4 Discriminative Phonotactics Dialect Recognition System

We saw in Chapter 5 that Arabic dialects significantly differ in terms of their phonotactic distribution. Particularly, we showed that the PRLM approach distinguishes Arabic dialects with good identification accuracy for the four broad Arabic dialects. In this section, we show how we can use the CD-phone classifiers described above to augment the phonotactic approach for Arabic dialect recognition.

Given an utterance U, we first run our CD-phone recognizer to obtain the most likely CD-phone sequence hypothesis along with the frame alignment. Then, for each CD-phone in the sequence, we extract its supervector and run the corresponding SVM classifier, as described in Section 8.2. We next attach the classification output to the CD-phone identity itself. If, for example, a CD-phone is [Voiced Cons.]–***r***–[Central Vowel] and the classification output is *Iraqi*, then we produce [Voiced Cons.]–***r***–[Central Vowel]$_{Iraqi}$. We apply this procedure to the entire CD-phone sequence. (See the sixth row in Figure 8.1.) We denote the output as the *annotated CD-phone sequence* (U_{text}). We thus transform the dialect recognition problem from classifying a speech utterance to classifying a textual sequence, similar to PRLM. Note that the idea of appending extra information to the phone identity is suggested by [Zissman, 1996], who attaches duration tags (Long/Short) to vowels based on their duration.

Now the task is classifying an annotated CD-phone sequence (U_{text}) to one of the dialects. One could simply adopt the PRLM approach using the annotated CD-phone sequences instead of raw phone sequences. Instead of applying a generative model (e.g., n-grams for each dialect), we train a discriminative classifier for each pair of dialects. These models are trained on the following list of *textual* features extracted from the annotated phone sequence:

- Frequency of annotated CD-Phone bigrams, e.g.,
  "[Nasal]–***r***–[Vowel]$_{Iraqi}$ [Voiced Cons.]–***a***–[Liquid]$_{Gulf}$"
- Frequency of bigrams with only one annotated CD-Phone, e.g.,
  "[Nasal]–***r***–[Vowel] [Voiced Cons.]–***a***–[Liquid]$_{Gulf}$"
- Frequency of annotated unigrams, e.g.,
  [!Central Vowel]–***E***–[Central Vowel]$_{Gulf}$
- Frequency of not annotated CD-Phone unigrams and bigrams, e.g.,

"[Nasal]–***r***–[Vowel] [Voiced Cons.]–***a***–[Liquid]"

- Frequency of context *independent* phone *trigrams*, e.g.,

  "s A l"

We normalize the feature vector by its euclidean norm to address durational differences across samples. Note that most of our features are annotated *CD-phone* unigrams and bigrams. This is because the classification is performed at the level of CD-phone — not context-independent (CI). Moreover, using CD bi-phones captures phonetic context better than CI bi-phones but less successfully than CI quad-phones. In fact, we have found that using a PRLM with bigram models trained on CD-phone sequences, instead of trigrams trained on CI phones, performs slightly better. The list of the features above is also ranked by feature importance, according to our experimental results.

There is a commonly held belief that discriminative classifiers are almost always to be preferred over generative classifiers due to modeling directly the posterior probability, or a map from input to class label. It has also been shown empirically that logistic regression and maximum entropy have typically lower asymptotic error than naive Bayes for multiple classification tasks as well as for text classification [Ng and Jordan, 2002]; [Nigam *et al.*, 1999]. Moreover, the advantage of using a discriminative classifier over an n-gram model in our case is due to the noisy identity tags attached to phones. An n-gram model trained on such sequences may not be robust; however a logistic classifier with a regularizer or SVM classifier will focus on the informative features and attempt to avoid irrelevant features that do not contribute to the classification task.

In addition, using a classification framework allows us to include different types of features at any level — even global features, which cannot be modeled using an n-gram model. In our experiments, we find that logistic regression with L_2-regularizer performs slightly worse than SVM with a linear kernel. However, surprisingly, logistic-regression with a L_2-regularizer typically performs slightly better than logistic regression with a L_1-regularizer, even though the L_1-regularizer is known for its feature selection capability [Ng, 2004]. For our detection task, we are interested in using confidence scores. Therefore, we choose logistic regression with a L_2-regularizer. We will make use of the posterior probability provided by logistic regression as our detection scores, described below.

8.5 Evaluation of Discriminative Phonotactics

To explain how we evaluate our Discriminative Phonotactics approach, recall that we train two types of models for each pair of dialects: the SVM CD-phone classifiers and a logistic regression classifier, which relies on features extracted in part from the predictions of the SVM classifiers. To train these two models, we divide our training speaker sets of Data II into two sets (SetI and SetII). For this approach, similar to the GMM-UBM experiments (in Chapter 7), we are also interested in evaluating it on 30s speech cuts. However, our training files are substantially longer. We therefore segment all files in both sets into approximately 30s-long cuts.

Now we first run the CD-phone recognizer on both sets to obtain a CD-phone sequence for each 30s cut. We then train the SVM CD-phone classifiers using SetI (see Section 8.2). Afterwards, we use these SVM classifiers to annotate the CD-phone sequences of SetII. Finally, we extract the textual features, described in Section 8.4, for each of these annotated sequences, producing one feature vector for each sequence. Using these vectors, we train a logistic regression classifier for each pair of dialects. One way to utilize the entire training data is to use the *second* set for training the SVM classifiers and the *first* set to train the logistic regression classifier. For classification, we use the average of the posteriors of both logistic classifiers; we term this a *cross training* method.

Recall that, during testing, given a trial r, we first run the CD-phone recognizer to obtain the most likely CD-phone sequence. We then extract a supervector for each CD-phone. Each supervector is classified using the corresponding SVM classifier to obtain a dialect label. Attaching the labels to the phones in the CD-phone sequence, we then extract our textual features to obtain a feature vector x_r. On the assumption that each trial is either a target dialect, D_t or a non-target D_{nt}, we use the posterior probability provided by the corresponding logistic regression model ($\Theta_{D_t D_{nt}}$) to represent our trial score: $p(D_t|x_r;\Theta_{D_t D_{nt}})$.

We use the NIST evaluation framework described in Section 4.5 to report our results. As shown in Figure 8.2, the Discriminative Phonotactics approach with the cross-training method, described above, yields an overall EER after pooling all test trials across dialect of 6.0%. The EER without cross-training is 6.9%. The discriminative approach outperforms both the standard GMM-UBM (15.3%) and GMM-UBM-fMLLR (11.0%).

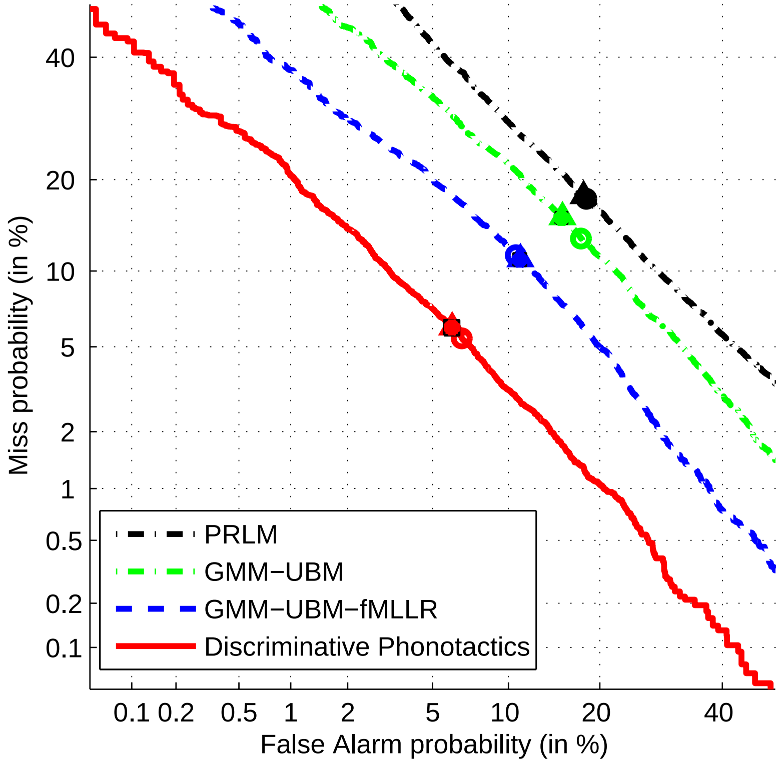


Figure 8.2: The overall DET curve for the four dialects with the best scoring scheme for each of the four approaches

8.6 Comparison to PRLM

As noted above, the PRLM approach is effective in identifying Arabic dialects. Moreover, since our Discriminative Phonotactics approach captures phonotactic features as well, we think it is essential to compare both using the same front-end, phone recognizer, and evaluation metric. For the PRLM experiment, every non-silent segment in the training data (of Data II) of all dialects is tokenized to the most likely CI phone sequence hypothesis, using the same CD-phone recognizer used for the Discriminative Phonotactics approach. Afterwords, using the CI phone sequences of dialect D_x, we train a phonotactic back-off trigram model with Witten-Bell smoothing for this dialect, denoted as λ_{D_x}, using the SRILM toolkit [Stolcke, 2002].

During testing, we calculate the scores as in (7.13), where $\mathcal{O}_r$ represents the most likely CI phone sequence of trial r, and $p(\mathcal{O}_r|\lambda_{D_x})$ represents the likelihood of $\mathcal{O}_r$ given the phonotactic trigram model λ_{D_x} of dialect D_x, and τ_r is the inverse of the number of phones in the sequence $\mathcal{O}_r$. Using our test data for the four dialects, and employing AllScoring scheme, the overall EER obtained by pooling the six pairs of dialects is 23.0%. When we use the target and non-target models only, i.e., using PairScoring, we achieve a significant improvement: the EER is 17.3%. The Discriminative Phonotactics approach achieves significantly higher results than PRLM, GMM-UBM, and GMM-UBM-fMLLR. See the DET curves for all approaches compared in Figure 8.2.

8.7 Evaluation per Dialect

We also compare the detection of each dialect against the rest separately, to determine whether the Discriminative Phonotactics approach outperforms the best baseline (GMM-UBM-fMLLR) in every dialect. As shown in Figure 8.3, we can see that, for all dialects, the Discriminative Phonotactics approach is superior when compared to this baseline.

In addition, we can see that the Egyptian dialect is the most distinguishable dialect across all dialects for both GMM-UBM-fMLLR and Discriminative Phonotactics. This is consistent with the results in Chapters 5 and 6. This could be due to several reasons: (1) According to linguists, and as observed in Chapter 6, the Egyptian Arabic has distinguishable linguistic cues (e.g., syllabic structure is simple); (2) our Egyptian dialect corpus contains mostly Cairene Arabic as opposed to the other dialect corpora which include multiple sub-dialects; (3) the Egyptian test corpus was not collected by the same company which collected the other three dialect corpora. Therefore, it is possible that different recording conditions have inflated the results. However, this is unlikely because our test utterances are from a completely different corpus than the training data.

We have also conducted more experiments in which we exclude the Egyptian dialect from our test trials. For the discriminative phonotactics approach, we obtain 10.5%; we

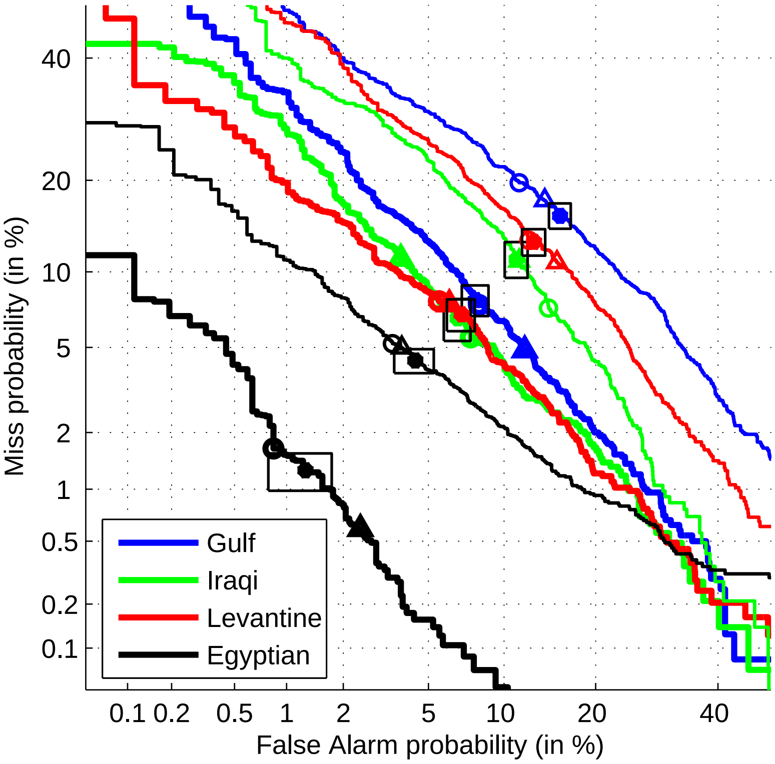


Figure 8.3: The DET curve of each dialect against the rest, comparing two approaches: Discriminative Phonotactics (thicker lines) vs. GMM-UBM-fMLLR

obtain 17.6% for the GMM-UBM with fMLLR adaptation; we obtain 23.1% for the GMM-UBM without adaptation, and 21.5% using the PRLM approach — all using PairScoring scheme.

8.8 Conclusions

In this chapter, we have introduced *Discriminative Phonotactics*, a novel approach to dialect recognition. We represent each CD-phone with a supervector, a result of stacking the mean vectors of MAP adapted GMMs of the CD-phone's acoustic model (HMM). Using this representation, we train an SVM classifier for each CD-phone type. We employ these classifiers to first *augment* the phonotactic sequences with dialect labels and then train a second discriminative classifier to classify dialects. Thus, Discriminative Phonotactics can be viewed as taking advantage of both phonotactic and acoustic-phonetic information.

Analyzing the performance of the Discriminative Phonotactics approach on detecting the four broad Arabic dialects, we have seen that it significantly outperforms PRLM and GMM-UBM baselines as well as our own improved version GMM-UBM-fMLLR (see Chapter 7). Discriminative Phonotactics achieves an EER of 6.0%, a reduction of 5% in EER (45.5% relative) over our best baseline (GMM-UBM-fMLLR).

An important use of this framework is its ability to automatically extract linguistic knowledge, specifically the phonetic cues that may distinguish one of our dialects from another. Particularly, the system can be used to distill which phones in which contexts are realized differently across dialects.

Chapter 9

Kernel-Based Method (Approach V)

9.1 Introduction

We have seen that the Discriminative Phonotactics approach is effective in recognizing Arabic dialects. An important aspect of this approach is its ability to automatically identify the subtle linguistics differences between dialects. In addition, this approach can be applied to *online* dialect identification, since we identify the dialect of single phones independently of future phones. To do online dialect identification, we will have to replace the back-end logistic classifier by a model that can work with a stream of data (e.g., an n-gram model).

On the other hand, there are some limitations of the Discriminative Phonotactics approach. We need to train a classifier for each CD-phone type for each pair of dialects. This can be quite expensive during training and recognition, and may be a little difficult to manage. Moreover, in this approach, the training speakers have to be split into two parts, one to train the SVM classifiers and another to train the logistic regression classifier to model the textual features. In this chapter, we introduce a *kernel-based* approach that allows us to train only a single SVM classifier for each pair of dialects. We design two main kernel functions to be used in the SVM classifier that computes the acoustic-phonetic similarities between pairs of utterances. We also experiment with two ways of extracting acoustic-phonetic features. Like the Discriminative Phonotactics approach, the kernel-based approach we propose here also relies on the hypothesis that dialects differ in their acoustic-phonetic structure. While we do not attempt to model the phonotactic distribution of dialects here, the approach implicitly captures CI or CD-phone unigram features.

9.2 Kernel-HMM (Using CD-Phone HMMs)

The feature extraction step in this approach and phonetic feature representation are the same as those in the Discriminative Phonotactics approach (see the steps in Section 8.2.1). In both, a supervector is extracted for each CD-phone using the means of the MAP adapted GMMs of the two or three state HMM of that CD-phone. Recall that every CD-phone HMM, which is an acoustic model of the phone recognizer, can be viewed as the UBM of this CD-phone type.

For the new approach, we represent each utterance U as a sequence S_U of tuples $(\vec{v}_i, \phi_i)$, such that $\vec{v}_i$ is the supervector of the i^{th} CD-phone in the sequence and ϕ_i is the identity of that CD-phone. Thus, an utterance U is represented as a sequence of tuples $S_U = \{(\vec{v}_1, \phi_1), (\vec{v}_2, \phi_2), ..., (\vec{v}_n, \phi_n)\}$, where n is the number of CD-phones in U. It should be noted that our representation retains the dependency between the phone identity and the supervector which 'summarizes' the spectral characteristics of the CD-phone. More importantly, as mentioned in Section 8.2.1, the supervector representation retains *some* of the phonetic structure of the CD-phone instance (albeit without the *complete* frame order). This is unlike the GMM-UBM approach which assumes that all frames are statistically independent (see Chapter 7). The sequence of tuples extraction procedure is illustrated in Figure 9.1.

9.2.1 Designing a Phone-Based SVM Kernel

From the sequences of tuples S_U produced for the utterances U of the training corpora, we next train an SVM classifier for each pair of dialects to distinguish one dialect from another at the utterance level as opposed to CD-phone level. To train an SVM classifier, we need a kernel function that computes the 'similarity' between pairs of training items. In our case, these items are sequences of tuples; therefore, we need to design a kernel

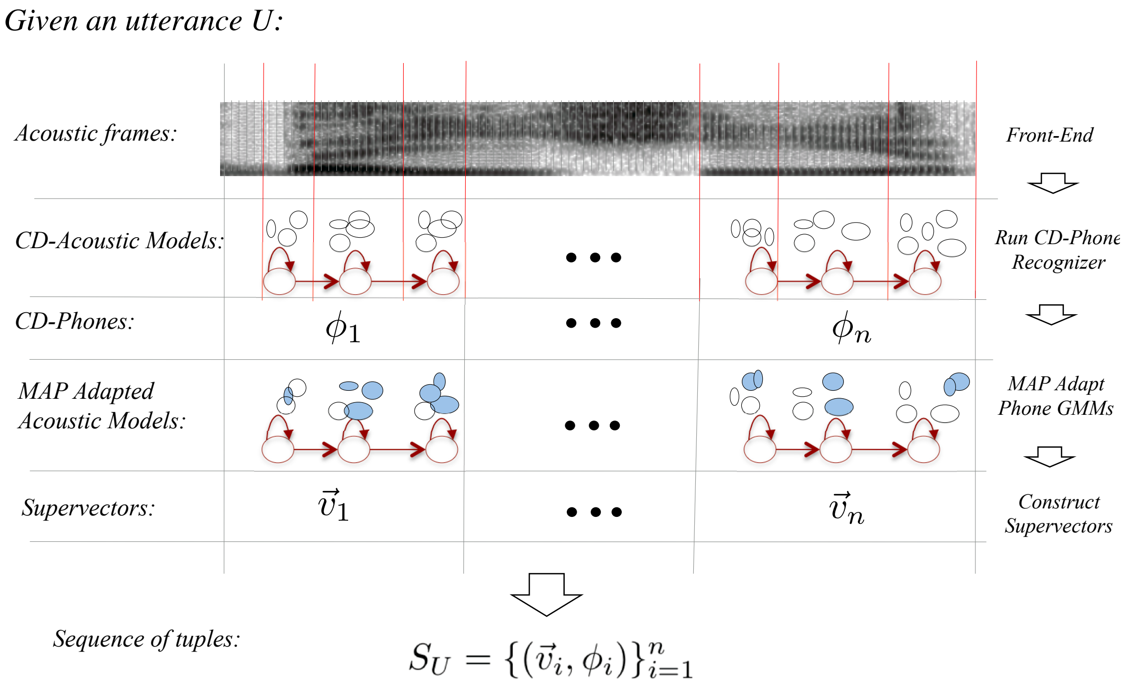


Figure 9.1: The Kernel-HMM sequence of tuples extraction procedure

function to compute the similarity between pairs of tuple sequences of utterances U_a and U_b. Let $S_{U_a} = \{(\vec{v}_i, \phi_i)\}_{i=1}^{n}$ and $S_{U_b} = \{(\vec{u}_j, \psi_j)\}_{j=1}^{m}$ be the tuple sequences of U_a and U_b, respectively. Our kernel function is defined in (9.1), where Φ is the phone inventory:

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} \sum_{i:\phi_i=\phi} \sum_{j:\psi_j=\phi} e^{-\|\vec{v}_i-\vec{u}_j\|^2/2\sigma^2} \tag{9.1}$$

This function computes the sum of RBF kernels between every pair of supervectors of CD-phone instances with the same type across the two utterances. It is straightforward to show that this kernel is positive definite, satisfying the Mercer condition. Note that this kernel ignores the order of supervectors in the sequence. As a result, phonotactic features (higher than unigram), for example, are not captured. Further research will be required to incorporate the sequential aspect in the kernel. We term this approach Kernel-HMM.

9.2.2 SVM Classification

We train an SVM classifier for each pair of our Arabic dialects as follows. We first extract the tuple sequences for all the 30s cuts, described in Section 8.5. Employing the kernel function above, we then compute a kernel matrix for each pair of dialects using the training 30s cuts. Next we train a standard binary SVM classifier for each pair of dialects using the pair's kernel matrix.[1] The regularization parameter C and σ (in the kernel function 9.1) are selected by 10-fold cross-validation on the training data. Since we need to recognize four Arabic dialects, we train a total of six binary classifiers.

$$f(S_U) = \sum_{i=1}^{N} \alpha_i y_i K(S_U, x_i) + b \tag{9.2}$$

During testing, we first run the phone recognizer to obtain the most likely phone sequence hypothesis for U along with the frame alignment for each CD-phone instance. We next extract the supervector for each CD-phone instance in the sequence, as described above, to obtain S_U. Using our kernel function, we then compute the kernel values $K(S_U, x_i)$, for all N support vectors x_i ($1 \leq i \leq N$). The final class prediction is then the sign of $f(S_U)$ in expression (9.2), where α_i and b are the estimated parameters of the dialect-pair SVM model (after training) and $y_i \in \{-1, 1\}$, the class label of support vector x_i.

9.2.3 Evaluation

For each pair of dialects, we use the SVM classifier described above to identify our test 30s utterance (U) to one of the dialects. To be able to plot a DET curve, we need confidence scores. We employ Wu et al. [Wu *et al.*, 2004]'s technique, implemented in LibSVM, which allows us to train SVM models that estimate posterior probabilities. Again, on the hypothesis that each trial is either a target dialect, D_t or non-target D_{nt}, we use the posterior probability provided by the corresponding SVM model ($\Theta_{D_t D_{nt}}$) to represent our trial score: $p(D_t|S_U; \Theta_{D_t D_{nt}})$. Using the same training/testing cuts in Data II, as we have done in the PRLM, GMM-UBM, GMM-UBM-fMLLR, and Discriminative Phonotactics, the overall EER obtained by pooling the six pairs of dialects is 5.88%, slightly better than Discriminative Phonotactics. While this difference is not statistically significant, the Kernel-HMM approach is more elegant and simpler to implement and easier to manage – we are required to train only one classifier for each pair of dialects. As shown in Figure 9.2, both Discrimi-

[1] In our implementation, we use LibSVM toolkit [Chang and Lin, 2001] to train our SVM models.

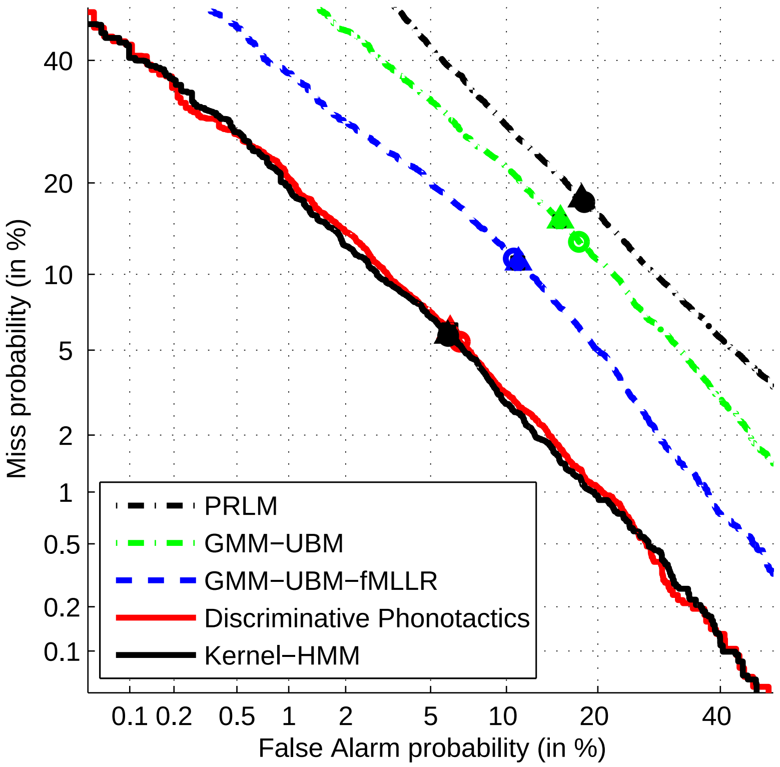


Figure 9.2: The overall DET curve for each of the five approaches

native Phonotactics and Kernel-HMM perform very similarly along all decision thresholds (both DET curves are very similar). We will see next how we can further improve the kernel-based approach.

9.3 Kernel-GMM (Using CI-Phone GMMs)

There are some limitations that may influence the performance of the Kernel-HMM approach:

1. Limiting the comparison between a pair of utterances to CD-phones of the same type may lead to a small number of comparisons overall, since the utterances may not share many CD-phones.

2. The number of frames for a given CD-phone is typically very small. Since our HMMs

are typically composed of three states, the number of frames aligned to each state will even be smaller. MAP adapting each of the three GMMs of the HMM using such a small number of frames may not lead to robust adapted-GMM parameter (i.e., mean) estimates.

3. We are limiting ourselves to the design and implementation of the phone recognizer, since we employ its acoustic models as the UBMs. For example, the number of CD-phones used must be small enough to be able to compare CD-phones of the same type. Also, the number of Gaussians per GMM in the phone recognizer must also be reasonably small; otherwise it is impractical to construct and maintain very high dimension supervector for every CD-phone.

We address these limitations by modifying the feature extraction step in Kernel-HMM approach: First, instead of comparing CD-phones across pairs of utterances, we can simply compare context-independent (CI) phones. Therefore, we extract supervectors from CI-phones as opposed to CD-phone. For our Arabic experiments, for example, our phone set will as a result consist of 34 CI-phones instead of 227 CD-phones. Since a pair of utterances will share more CI-phones than CD-phones, more phones will be compared to each other. This change will resolve the first problem above. To alleviate the second problem, instead of using an HMM to represent the UBM for each CD-phone, we can simply train a single GMM-UBM for each CI-phone type. Note that, as opposed to HMMs, GMMs will not capture structural information in phones (i.e., sub-phonetic information: beginning/middle/end of phones). The third problem is resolved simply by not employing the acoustic models of the phone-recognizer as our UBMs. As noted, we can instead train a GMM-UBM for each CI-phone from scratch, as we describe next. We term this new approach Kernel-GMM.

9.3.1 Phone GMM-UBM

The first stage in the Kernel-GMM approach, after front-end pre-processing, is to use our phone recognizer to obtain the most likely CI-phone sequence hypothesis for each utterance in the training corpora. We then extract the PLP feature vectors for each frame of each phone instance in the sequence. Note that these features are also extracted after normalization (CMVN) and fMLLR transformation. We next train a GMM-UBM for each CI-phone type using all frames of all instances of that phone type from all dialects. We denote this GMM-UBM as *phone GMM-UBM*. All GMMs are ML-trained, with 100 Gaussian components, using the EM algorithm.[2] To avoid a bias for any particular dialect in the GMM-UBM, we select an equal number of frames from each dialect for each phone GMM-UBM. From our 34 MSA phone inventory we thus train 34 phone GMM-UBMs.

9.3.2 Creating Phone-GMM-Supervectors

For the Kernel-GMM approach, we extract acoustic-phonetic features from *CI*-phones for a given utterance U. Using U's phone hypothesese and frame alignments, we represent U with a sequence S_U of tuples $(\vec{v}_i, \phi_i)$, such that $\vec{v}_i$ is the supervector of the i^{th} *CI*-phone instance in the sequence and ϕ_i is the identity of that CI-phone. The supervector $\vec{v}_i$ of phone ϕ_i is constructed as follows. We use the acoustic frames aligned to ϕ_i to MAP adapt the Gaussian means of the corresponding phone GMM-UBM, with a relevance factor of $r = 0.1$. We denote the resulting GMM as the *adapted phone-GMM*. The supervector $\vec{v}_i$ is the result of stacking all the mean vectors of the Gaussians of this adapted phone-GMM. We also include the duration of ϕ_i as an additional feature in $\vec{v}_i$. The duration feature here is computed as the log of the number of frames in the phone. The extraction procedure of the sequence of tuples is illustrated in Figure 9.3.

9.3.3 Evaluation

Extracting the sequences of tuples described above, we compute our kernel matrix using the kernel function (9.1) for each pair of dialects. We then train a binary SVM classifier for each pair of dialects using the same training data used in the Kernel-HMM approach. We compute the overall DET curve on the same test data as well. As shown in Figure 9.4, Kernel-GMM yields a significant improvement in EER (4.9%) over Discriminative Phonotactics and Kernel-HMM approaches, a relative reduction in EER of 18.3% over Kernel-HMM. We can conclude from these results that, with our kernel function, modeling phonetic differences using GMM supervectors of CI-phones is more robust than modeling them with

[2] The number of Gaussians employed for each phone type was determined empirically.

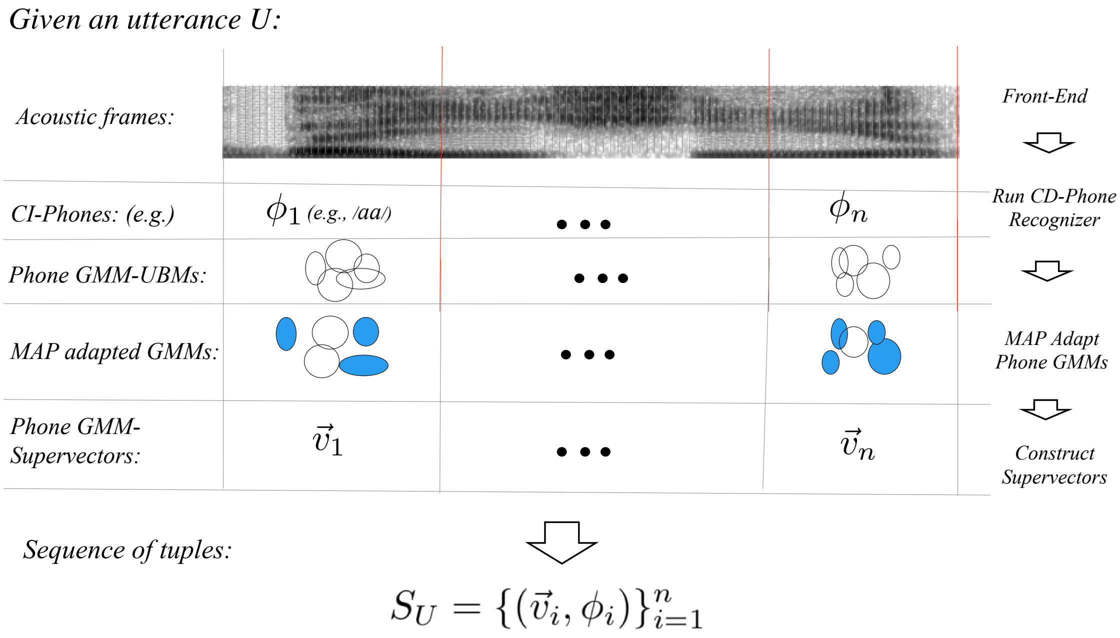


Figure 9.3: The Kernel-GMM sequence of tuples extraction procedure

HMM supervectors of CD-phones.

9.3.4 Time Complexity

Let us turn our attention to calculating the time complexity of computing the kernel function and matrix in this approach for training and testing. Let p_k^i and m_k^j be the number of instances of phone type k (where, $1 \leq k \leq |\Phi|$, and Φ is the phone inventory) in utterances U_i and U_j, respectively. For simplicity, let assume that all supervectors of different phone types have the same size (D), which is the case in Kernel-GMM ($D = 4,000$). Denoting $M = max_{i,k}\{m_k^i\}$, and assuming that the Euclidian distance between two D-dimensional vectors takes $\mathcal{O}(D)$, the time complexity of comparing a pair of utterances using the kernel function (9.1) is upper bounded with $\mathcal{O}(M^2|\Phi|D)$ (from $\mathcal{O}(\sum_{k=1}^{|\Phi|} p_k^j m_k^i D)$). For example, the average of the frequency of the most frequent phone $\backslash a \backslash$ in our Arabic data across all utterances is about 53. Therefore, the expected number of comparisons for this phone is 53^2 for each pair of utterances.

To construct the kernel matrix for training, we have to compute the kernel function between each pair of the N training utterances. Thus, it takes $\mathcal{O}(N^2M^2|\Phi|D)$ (from $\mathcal{O}(\sum_{i=1}^{N}\sum_{j=i}^{N} M^2|\Phi|D)$). For testing, we have to compute this kernel function with all

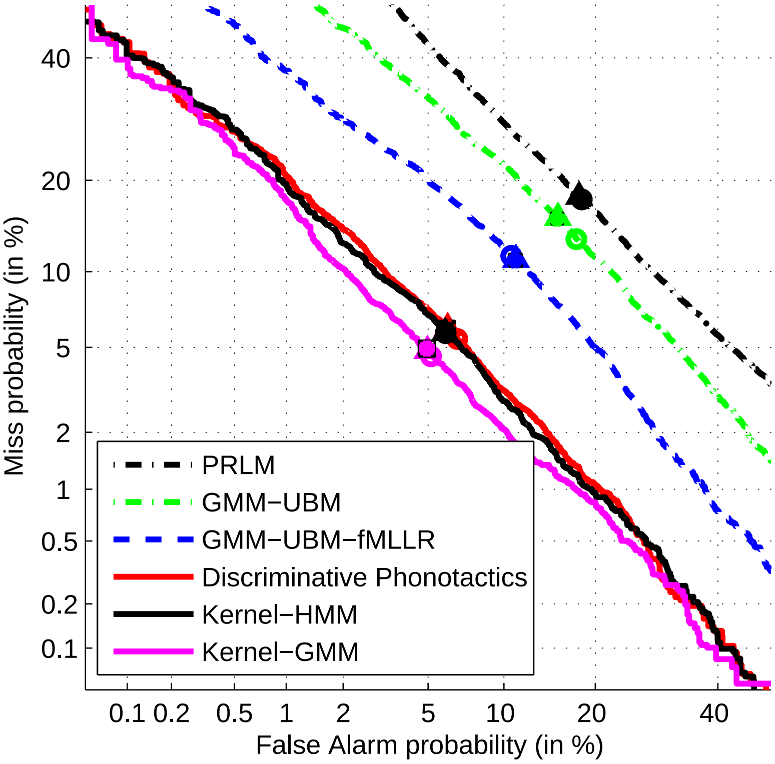


Figure 9.4: The overall DET curve for each of the six approaches

support vectors x_l ($1 \leq l \leq L$). Thus, the time complexity for testing is $\mathcal{O}(M^2L|\Phi|D)$ (from $\mathcal{O}(\sum_{l=1}^{L}\sum_{k=1}^{|\Phi|} x_k^l u_l D)$, where u_l is the number of instances of phone type l in the input utterance U).

We can see that computing the kernel function in (9.1) is quite expensive, partly due to the cross-comparison between every phone *instance* of the same type across each pair of utterances. This becomes increasingly significant when the training/testing utterances are long (leading to large M). Note that a smaller phone inventory would result in a larger number of instances of each type within each utterance; because the cost is linear in inventory size, but quadratic in instance count, this would increase cost. Another significant disadvantage of the Kernel-GMM-Instance approach is that since each phone instance typically consists of just a few frames, performing the MAP adaptation at this level leads to robustness issues with the parameter estimates for the adapted phone-GMM.

9.4 Kernel-GMM-Type

As we saw in the previous section, modifying the feature extraction step from CD-HMM supervectors to CI-GMM supervecors led to a significant improvement over all other approaches. However, as we also noted, the time complexity of the Kernel-GMM approach can still be an obstacle. In this section, we propose a new kernel function that can be computed substantially faster than the function in (9.1). Instead of comparing supervectors of phone *instances*, we will compare the supervectors of phone *types*. Therefore, we will have a constant number (as many as phone types) of comparisons between a pair of utterances.

9.4.1 Creating Phone-Type-GMM-Supervectors

We create one supervector for each phone type in a given utterance U. Similar to Kernel-GMM, we first run the phone recognizer to obtain the most likely CI-phone sequence hypothesis for U along with the frame alignment for each phone instance. Instead of performing MAP adaptation for each phone instance (as for Kernel-GMM), here we use all the frames of all the phone instances of the same phone type in U to MAP adapt the corresponding phone GMM-UBM. Thus, we obtain $|\Phi|$ adapted phone-GMMs.[3] Again, we adapt only the means of the Gaussians using a relevance factor of $r = 0.1$. The adapted GMM means are then stacked to construct a supervector for each phone type. This representation captures the 'general' realization of each phone type as opposed to the individual realization of each phone instance, as in Kernel-GMM. We term this approach as Kernel-GMM-Type.

9.4.2 Designing a Phone-Type-Based SVM Kernel

An utterance U is represented by a set S_U of supervectors, each supervector corresponding to one phone type. Therefore, the size of S_U is at most the size of the phone inventory ($|\Phi|$).[4] We denote the supervector $\vec{u}$ of phone type ϕ, as $\vec{u}_\phi$. Let $S_{U_a} = \{\vec{u}_\phi\}_{\phi\in\Phi}$ and $S_{U_b} = \{\vec{v}_\phi\}_{\phi\in\Phi}$ be the phone-type supervector sets of utterances U_a and U_b, respectively. Our new kernel function is defined in (9.3). It compares the general phonetic realization of

[3] Recall that we denote our phone inventory as Φ – so, the number of phone types is $|\Phi|$.

[4] Since we model 34 MSA phones in the Arabic phone recognizer, $|\Phi| = 34$.

the same phone types across a pair of utterances, as opposed to the realization of every pair of individual-phone instances of the same type across the pair of utterances, as in (9.1). We term this approach Kernel-GMM-Type.

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} e^{-\|\vec{u}_\phi - \vec{v}_\phi\|^2/2\sigma^2} \tag{9.3}$$

9.4.3 Evaluation

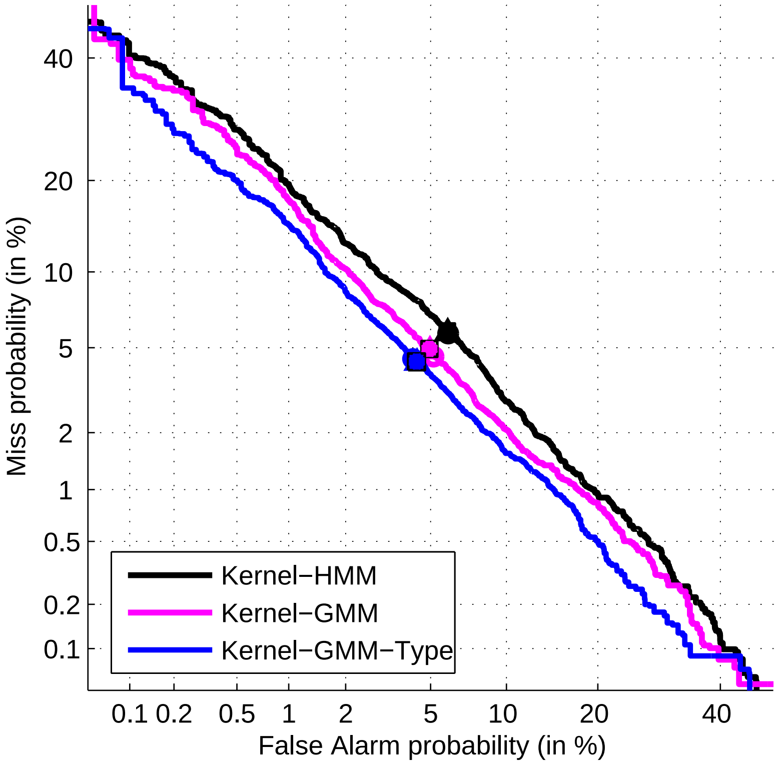


Figure 9.5: Comparing the overall DET curves of the kernel-based approaches

Similar to the Kernel-GMM evaluation, we compute a kernel matrix but use the kernel function in (9.3) for each pair of dialects. We then train a binary SVM classifier for each pair of dialects using the same training data used in Kernel-GMM. We compare the performance of the three kernel approaches in Figure 9.5. Interestingly, the Kernel-GMM-Type approach performs better than Kernel-GMM. The EER of kernel-GMM-Type is 4.35%. Nonetheless, the improvement in ERR is not statistically significant. Although the main motivation for Kernel-GMM-Type is to improve the time complexity of Kernel-GMM, we also achieve a reduction in EER. We believe that this is due to the resolution of the second problem in Section 9.3 (i.e., having insufficient number of frames to MAP adapt the GMMs). Recall in Kernel-GMM-Type, *all* the frames of the same phone type in an utterance are used to MAP adapt the corresponding phone-type GMM. Thus, the parameters of the adapted phone-type GMMs are more robustly estimated than those of adapted phone-GMMs that utilize the frames of individual phone instances, as in Kernel-HMM and Kernel-GMM.

9.4.4 Kernel Choice

The kernel functions we have designed thus far are the sum of RBF kernels between phone-type supervectors. In this section, we experiment with replacing the RBF kernel by: (1) a linear kernel and (2) an upper-bound of Kullback–Leibler (KL) divergence.

9.4.4.1 Linear Kernel

Employing the linear kernel, we define our kernel function between a pair of utterances in (9.4). We simply replace the RBF kernel in (9.3) with a linear kernel. Although the RBF kernel is more powerful than the linear kernel, particularly in modeling linearly inseparable data, the linear kernel typically performs very well in high-dimensional data. One of the advantages of employing a linear kernel over RBF is that we need not determine/tune the parameter σ in the expression (9.3).

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} \vec{u}_{\phi}^{t} \cdot \vec{v}_{\phi} \tag{9.4}$$

We evaluate the linear-based kernel function in the expression (9.4) using the same settings as in Kernel-GMM-Type. As shown in Figure 9.5, when applying the linear-based kernel, the EER is 4.45% which is slightly worse, but not significantly, than the EER when using the RBF kernel.

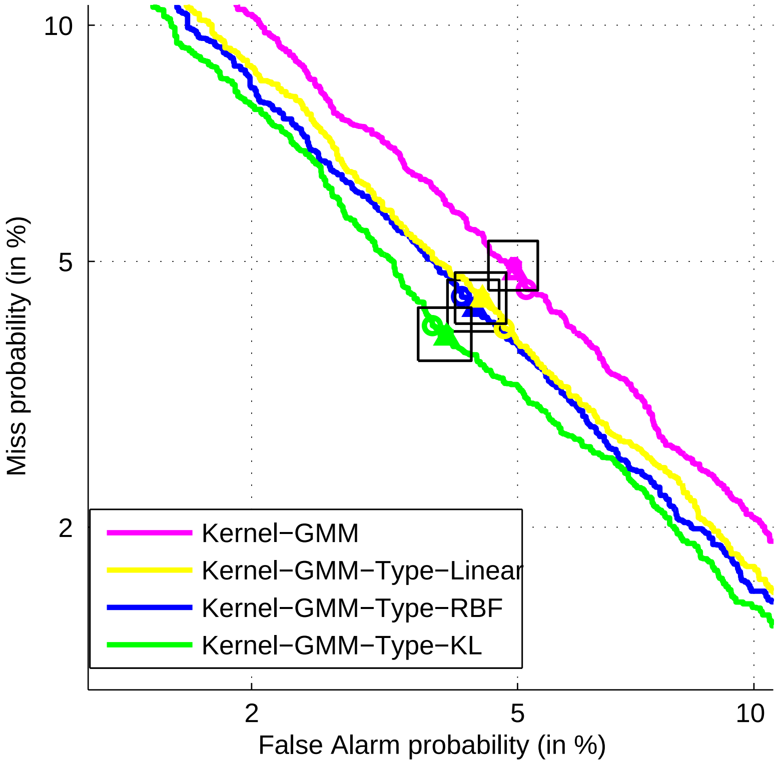


Figure 9.6: Comparing different kernels for Kernel-GMM

9.4.4.2 KL-Divergence-based Kernel

Recall that a supervector in the Kernel-GMM-Type approach is a result of stacking the Gaussian means of a MAP adapted phone-type GMMs. Instead of comparing the means of the GMMs in our kernel functions, we can compare the KL-divergence between the two adapted GMMs. Unfortunately, the KL divergence is not symmetric and does not satisfy the Mercer condition and thus cannot be straightforwardly used in SVM.

Moreno et al. [2004] have shown how to create a kernel function between two GMMs by exponentiating the negative symmetric KL-divergence value along with scaling and shifting it. Since there is no closed form for computing the KL-divergence between GMMs, the authors have resorted to applying Monte Carlo and approximation methods. However, these solutions complicate the approach. Do [2003] on the other hand has shown that, using the log-sum inequality, the KL-divergence between two GMMs g_a and g_b ($KL(g_a \parallel g_b)$) is upper bounded as in (9.5), where ω_i^X, μ_i^X, Σ_i^X are the weight, Gaussian mean vector, covariance matrix of Gaussian i in GMM X, respectively. d is the space dimension. The symmetric version of the KL divergence is shown in (9.6).

$$\begin{aligned} KL(g_a \parallel g_b) &\leq KL(\omega^a \parallel \omega^b) + \sum_i \omega_i^a KL(\mathcal{N}(.;\mu_i^a,\Sigma_i^a) \parallel \mathcal{N}(.;\mu_i^b,\Sigma_i^b)) \quad (9.5)\\ &= KL(\omega^a \parallel \omega^b)\\ &\quad + \sum_i \frac{\omega_i^a}{2}[log(\frac{|\Sigma_i^b|}{|\Sigma_i^a|}) + Tr(\Sigma_i^{b^{-1}}\Sigma_i^a) - d + (\mu_i^a - \mu_i^b)^t \Sigma_i^{b^{-1}}(\mu_i^a - \mu_i^b)]\\ &\doteq D(g_a \parallel g_b) \end{aligned}$$

$$\begin{aligned} KL^{sym}(g_a \parallel g_b) &= KL(g_a \parallel g_b) + KL(g_b \parallel g_a) \quad (9.6)\\ &\leq D(g_a \parallel g_b) + D(g_b \parallel g_a) \quad (9.7)\\ &\doteq D^{sym}(g_a \parallel g_b) \end{aligned}$$

$$\begin{aligned} D^{sym}(g_a \parallel g_b) &= \sum_i \omega_{\phi,i}(\mu_i^a - \mu_i^b)^t \Sigma_{\phi,i}^{-1}(\mu_i^a - \mu_i^b) \quad (9.8)\\ &= K_\phi(\mu^a,\mu^a) - 2K_\phi(\mu^a,\mu^b) + K_\phi(\mu^b,\mu^b) \quad (9.9) \end{aligned}$$

where,

$$\begin{aligned} K_\phi(\mu^a,\mu^b) &= \sum_i \omega_{\phi,i}\mu_i^a \Sigma_{\phi,i}^{-1}\mu_i^b \quad (9.10)\\ &= \sum_i (\sqrt{\omega_{\phi,i}}\Sigma_{\phi,i}^{-\frac{1}{2}}\mu_i^a)^t(\sqrt{\omega_{\phi,i}}\Sigma_{\phi,i}^{-\frac{1}{2}}\mu_i^b) \quad (9.11) \end{aligned}$$

Since we only MAP adapt the means of the phone GMM-UBMs, the covariance matrices and weight vectors (ω^a and ω^b) of the adapted phone-GMMs are the same as those of the corresponding phone GMM-UBM, for all utterances. Let $\omega_{\phi,i}$ and $\Sigma_{\phi,i}$ respectively be the weight and covariance matrix of Gaussian i of the phone GMM-UBM of phone-type ϕ; and let assume diagonal covariance matrices (which is in fact the case for our GMMs). From

(9.7), the approximation of the symmetric KL-divergence between two adapted phone-GMMs is therefore equivalent to the expression in (9.8). Using the distance metric in (9.8), Campbell el al. [2006b] have found a corresponding kernel function between the two GMM mean vectors (supervectors μ^a and μ^b), as shown in the expressions (9.10) and (9.11). Note this kernel is the sum of a dot product between scaled (by $\sqrt{\omega_\phi}\Sigma_\phi^{-\frac{1}{2}}$) GMM mean vectors. As a result it satisfies the Mercer condition. Note also that this scaling can be pre-computed for each mean vector before computing the kernel matrix.

Using the KL-divergence-based kernel function (9.11), we define our new kernel function between a pair of utterances:

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} K_\phi(\vec{u}_\phi - \vec{\mu}_\phi, \vec{v}_\phi - \vec{\mu}_\phi) \tag{9.12}$$

where μ_ϕ is the stacked mean vectors of the phone-GMM-UBM of phone-type ϕ. The subtraction of $\vec{\mu}_\phi$ in (9.12) from the supervectors is to allow zero contributions from Gaussians that are not affected by the MAP adaptation, which will result in sparse supervectrors. We have observed that this subtraction slightly improves ERR. We term this approach Kernel-GMM-Type-KL.

It is interesting to note that for a linear kernel K_ϕ such as (9.11), we can represent each utterance S_{U_x} in (9.12) with a single vector. This vector, say W_x, is formed by stacking the phone-type supervectors (after scaling by $\sqrt{\omega_\phi}\Sigma_\phi^{-\frac{1}{2}}$ and subtracting the corresponding $\vec{\mu}_\phi$) in some (arbitrary) fixed order, with zero supervectors for phone types not in U_x. This representation allows the kernel in (9.12) to be written as:

$$K(S_{U_a}, S_{U_b}) = W_a^T W_b \tag{9.13}$$

The vector-of-supervectors W_x can be viewed as the ‘phonetic fingerprint’ of the utterance. We hypothesize that such a representation can be useful for multiple speech applications, including speaker verification and identification. It is important to note that, in our vector-of-supervectors, the phone labels constrain which Gaussians can be affected by the MAP adaptation, i.e., the comparison incorporates the linguistic constraints realized by the phone recognizer. This is in contrast to the GMM-supervector representation [Campbell *et al.*, 2006a] for which, in theory, any Gaussian in the GMM-UBM can be affected by any frame of any phone – ignoring the linguistic context of each frame.

This representation is also interesting because, given that an utterance can be represented in one vector, we can experiment with a different classifier instead of SVM. In fact, we have evaluated our American English Southern vs. Non-Southern using a logistic regression with L_2 regularizer. Unsurprisingly, due to the close relationships of these classifiers, the logistic regression classifier performs slightly but not significantly better than the SVM classifier which uses the kernel in (9.12) [Vapnik, 1999].

9.4.5 Evaluation

We evaluate Kernel-GMM-Type-KL using the same data sets and settings as for Kernel-GMM. As shown in Figure 9.6, this approach provides us with our best results: an EER of 3.96% with significant difference from Kernel-GMM (19.8% relative improvement). Table 9.1 and 9.2 respectively present the classification accuracy and EER for each pair of dialects. The pairwise DET curves are also shown in Figure 9.7.

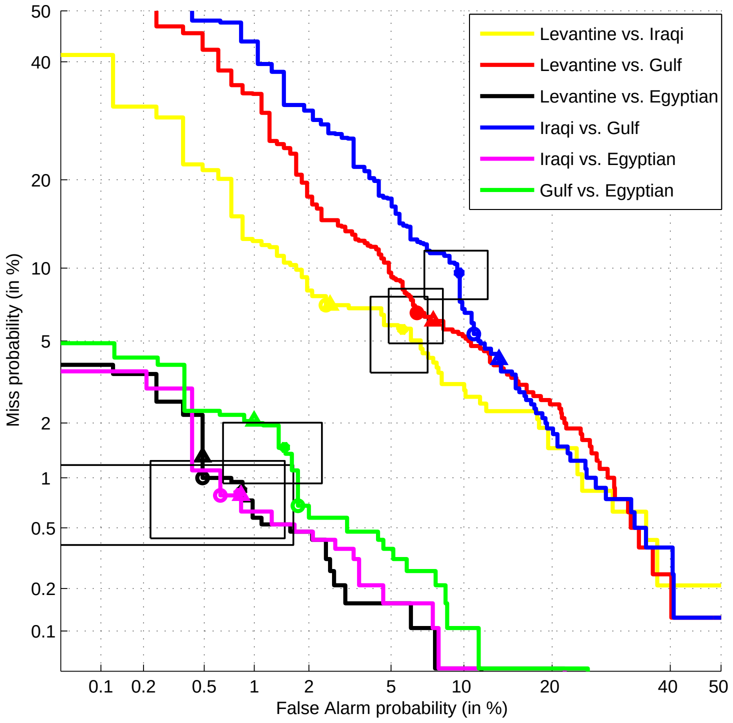


Figure 9.7: DET curves for each pair of dialects

We next compare the performance of Kernel-GMM-Type-KL to that of GMM-SVM [Campbell *et al.*, 2006b] with the same front-end. For the GMM-SVM approach, we make use of our GMM-UBM-fMLLR, described in Chapter 7. Recall that we showed in Chapter 7 that transforming features using fMLLR significantly improves results for GMM-UBM. Recall also that this UBM is composed of 2048 Gaussians. We use here a relevance factor r=16 to adapt the UBM given an input utterance. We term this approach GMM-fMLLR-SVM. This approach can be viewed as a specific case of our Kernel-GMM-Type-KL, since in GMM-fMLLR-SVM we ignore all phone labels by treating all of them as a *single* general phone.[5]

We compare the classification accuracy and EER of our Kernel-GMM-Type-KL to that of GMM-fMLLR-SVM for each pair of dialects in Tables 9.1 and 9.2. The DET curves comparing the two approaches are shown in Figure 9.8. We can see that Kernel-GMM-Typle-KL yields better results for all dialect pairs. These results indicate that modeling/comparing phone-type supervectors is better than simply comparing supervectors at the whole utterance level, as proposed in [Campbell *et al.*, 2006b]. As discussed above, in Kernel-GMM-Type-KL, the comparison between utterances incorporates linguistic constraints which allow only certain Gaussians to be affected by the MAP adaptation.

There is another significant advantage of Kernel-GMM-Type-KL over GMM-SVM, which is an advantage in running time. In Kernel-GMM-Type-KL, we MAP adapt each individual phone type modeled with a relatively small number of Gaussians, using a relatively small number of frames. Let n_i be the number of Gaussians of phone-GMM i (where $1 \leq i \leq |\Phi|$) and t_i is the number of frames for this phone in a given utterance; then the time complexity of MAP adapting all the phone GMM-UBMs in the utterance is $\mathcal{O}(\sum_i n_i t_i)$. Assuming that we have the same total number of Gaussians for the single UBM for GMM-SVM ($\sum_i n_i$), the time complexity of MAP adapting the UBM using all frames of this utterance in contrast is $\mathcal{O}(\sum_i n_i \sum_i t_i)$, which is substantially higher than MAP adapting each individual phone GMM-UBM. Note also that MAP adapting individual Gaussians can be easily and more efficiently parallelized. Nevertheless, unlike Kernel-GMM-Type-KL, GMM-SVM does not require phone recognition unless fMLLR transform is used.

[5] We still use the KL-divergence-based kernel and mean shifting.

We observe here, similar to the results of all our approaches, that the Egyptian dialect is the easiest to classify of all the dialects, followed by Levantine (See our discussion in Section 8.7.). Also, we see that Iraqi/Gulf is the most difficult pair of dialects to distinguish (see the discussion in Section 5.5.1).

Dialect Pair	Kernel-GMM-Type-KL	GMM-fMLLR-SVM
Egyptian/Gulf	99.3	95.9
Egyptian/Iraqi	99.2	98.1
Egyptian/Levantine	98.9	97.3
Levantine/Iraqi	95.8	91.3
Levantine/Gulf	93.3	88.9
Iraqi/Gulf	92.3	86.7

Table 9.1: Comparing the classification accuracy (in %) of our approach to GMM-fMLLR-SVM for each pair of dialects

Dialect Pair	Kernel-GMM-Type-KL EER (%)	GMM-fMLLR-SVM EER (%)
Egyptian/Gulf	1.5	3.0
Egyptian/Iraqi	0.8	1.9
Egyptian/Levantine	0.9	2.5
Levantine/Iraqi	5.6	9.2
Levantine/Gulf	6.6	10.6
Iraqi/Gulf	9.6	13.8

Table 9.2: Comparing the EER of our approach to GMM-fMLLR-SVM for each pair of dialects

9.4.6 Time Complexity

Not only do we obtain the best results with Kernel-GMM-Type-KL, but also the time complexity of this approach is also substantially lower than that of Kernel-GMM. Assuming

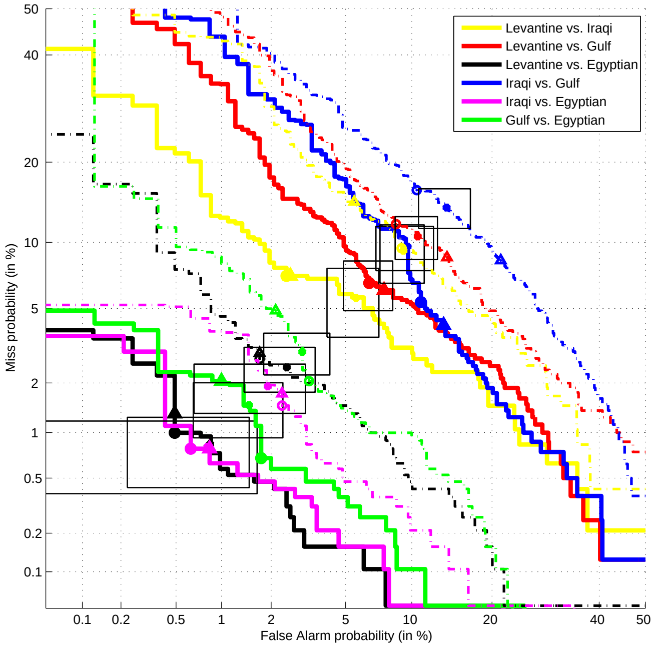


Figure 9.8: Comparing the DET curves of Kernel-GMM-Type-KL to GMM-fMLLR-SVM for each pair of dialects; Dotted thin lines are GMM-fMLLR-SVM

that all utterances have at least one instance from each phone type, the time complexity of computing the kernel function (9.12) on a pair of utterances is $\mathcal{O}(|\Phi|D)$, where D is the GMM supervector size; for the Arabic experiments, $D = 4,000$; $|\Phi|$ is the size of the phone inventory, which is a small constant.[6] Thus, constructing the kernel matrix for N utterances is $\mathcal{O}(N^2D)$. Note that, unlike Kernel-GMM, for which the complexity is $\mathcal{O}(N^2M^2D)$, the time complexity here is independent of the duration of utterances, since we compare phone types as opposed to phone instances. We observed run-time speed improvements of about 12-15×.

[6] We assume that the dot product between two D-dimensional vectors takes $\mathcal{O}(D)$.

9.5 Biphones, Bi-Manner of Articulation and CD-Phones

All the features employed in our kernel-based approaches are mono-phonetic-based features – i.e., supervectors extracted from frames of individual phones. We attempt in this section to model differences of larger context, features extracted from biphones (two consecutive phones) employing Kernel-GMM-Type-KL. We conduct a series of experiments to test whether these types of features improve classification accuracy. Since our kernel-based approaches and Discriminative Phonotactics perform very well on the Egyptian dialect (classification accuracy of about 99%), we decided to test the approaches proposed in this section on only the three other Arabic dialects (Levantine, Gulf, and Iraqi). This allows us to conduct more experiments, since now we need to train only three binary classifiers as opposed to six. In this section, we report classification accuracy for each of the three binary classifiers.

9.5.1 Adding Biphone Features

Augmenting the mono-phonetic features with biphonetic, we first train a phone GMM-UBM for each phone type, exactly as in Kernel-GMM-Type. We then train a GMM-UBM for some biphones using the frames aligned to the biphone (from both phones) pooled from all training data. Since the number of possible biphones is quite large (i.e., $|\Phi|^2$), we select only a subset of biphones. In our experiments we choose the 100 most frequent biphones. Then we employ the steps in Kernel-GMM-Type-KL to compute the kernel matrix followed by SVM classification. This approach can be viewed as augmenting Φ to include not only the phone inventory but also the 100 most frequent biphones.

We test the effect of adding such biphonetic features to our framework on classifying Iraqi/Gulf. The accuracy of using the Kernel-GMM-Type-KL with mono-phonetic features only, as in Section 9.4.4.2, is 92.3%. When we add the biphonetic features, we achieve a slight improvement: 92.5%; the improvement is not statistically significant. Note that this approach relies heavily on the correctness of the phone recognizer; biphones will be included in the features only if both were correctly recognized. This could explain the lack of significant improvement.

9.5.2 Adding Bi-Manner of Articulation Features

As noted above, adding all possible biphones can increase the complexity of the algorithm. Instead of limiting the number of biphones, as proposed in the previous section, we decided to cluster consonants based on their manner of articulation and to cluster the six MSA vowels in one category, as shown in Table 9.3. Instead of extracting features from biphones, as described above, we now extract features from two consecutive manner-of-articulation-based clusters, which we term *Bi-Manner of Articulation* (BMA). For example, we may extract a supervector for ‘a nasal followed by a vowel’, and a different supervector for ‘a nasal followed by an unvoiced plosive’.

We have 7 categories; therefore, we build 7^2 (= 49) GMM-UBMs, one for each BMA, similar to the biphones in the previous section. Using the Kernel-GMM-Type-KL framework and augmenting the mono-phonetic features with these BMA features, we obtain a slight improvement approaching significance (p-value = 0.075) for Iraqi/Gulf, but no significant improvement for the other two dialect pairs (see Table 9.4.) Note, in this table, that the BMA features do not perform well alone. As future research, we hypothesize that biphones should be automatically clustered based on their ability to discriminate dialects instead of manner of articulation.

Category:	Phones
Nasal	m, n
Unvoiced Plosives	t, k, T, q, G
Voiced Plosives	b, d, D, j
Unvoiced Fricatives	f, v, s, S, $, x, H, h
Voiced Fricatives	V, z, Z, g, E
Approximants and Trill	w, y, l, r
Vowels	a, A, i, I, u, U

Table 9.3: Categorizing phones based on manner of articulation

Dialect Pair	Mono-phones	BMA	Mono-phones + BMA	CD-Phones
Levantine/Iraqi	95.8	94.4	95.3	93.1
Levantine/Gulf	93.3	92.8	93.6	92.3
Iraqi/Gulf	92.3	91.7	93.4	91.5

Table 9.4: The classification accuracy (in %) with different feature types

9.5.3 CD-Phones with GMMs

We have seen in Section 9.4 that modeling CI-phone types with GMMs performs better than modeling CD-phone instances with HMMs. We think it is valuable to test whether modeling CD-phone *types* with *GMMs* improves results or not. To do that, we simply replace Φ, the set of 34 phones, by the set of 227 CD-phones, described in Section 7.7. Training a GMM-UBM for each of these CD-phones, we test our Kernel-GMM-Type-KL approach using this phonetic set. Interestingly, as shown in Table 9.4, this approach achieves the worst results, significantly worse for the Levantine/Iraqi classifier. We hypothesize that the reason for this poor performance is that, as the phone inventory size increases, fewer phones are compared to each other across pairs of utterances. We discuss this problem in Section 9.3 (Problem 1).

9.6 Comparison to a State-of-the-Art Approach

Using a state-of-the-art system, Torres-Carrasquillo et al. [Torres-Carrasquillo *et al.*, 2008] showed that GMM-UBM-based models discriminatively trained with SDC features with an eigen-channel compensation component and VTLN and with a back-end classifier achieve an EER of 7.0% on three Arabic dialects (Gulf, Iraqi, and Levantine) using the same Appen corpora employed here. (See Section 4.2 for more details) To compare our performance to this work, we experiment with Kernel-GMM-Type-KL (with monophones), using both the training *and* the development data used by [Torres-Carrasquillo *et al.*, 2008] to train our SVM models; we evaluate on the test cuts used in [Torres-Carrasquillo *et al.*, 2008].[7]

[7] We thank P. Torres-Carrasquillo and N. Chen for providing us with the segmentations.

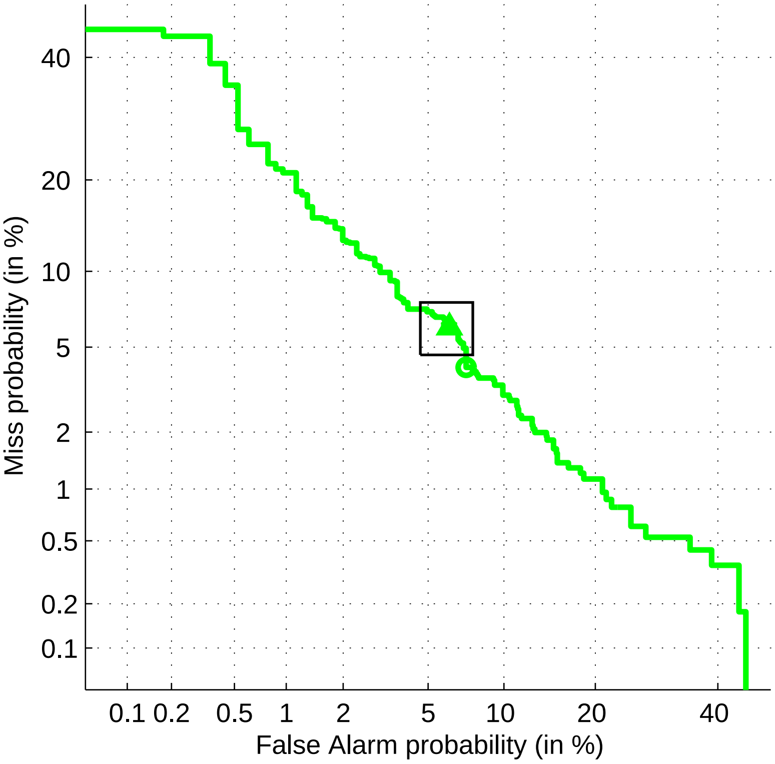


Figure 9.9: The overall DET curve of Kernel-GMM-Type-KL on test cuts on three Arabic dialects using Torres-Carrasquillo et al. [2008]'s test cuts

Using this data segmentation, our Kernel-GMM-Type-KL approach achieves better performance than [Torres-Carrasquillo *et al.*, 2008]: an EER of 6.1% (12.9% relative improvement in EER) – see Figure 9.9. Moreover, testing our Kernel-GMM approach on these segments, we obtain an EER of 6.4%. Our results suggest that Kernel-GMM-Type-KL has considerable potential, particularly when VTLN and channel compensation components are added. According to [Torres-Carrasquillo *et al.*, 2008], these components reduce the EER from 18% to 12% using the GMM-UBM approach on this task. Note that we obtain higher EER on these three dialects than our overall EER for all four dialects since Egyptian Arabic is the most distinguishable dialect of the four. In the next chapter we also compare our system to Torres-Carrasquillo et al [2008]'s system on American English vs. Indian English.

9.7 Conclusions

In this chapter, we have introduced another novel approach for dialect recognition, based on the hypothesis that some phones are realized quite differently across dialects. Given an input utterance, we employ a phone recognizer to obtain the most likely phone sequence. We extract GMM supervectors for each phone instance/types in the sequence. Using these supervectors, together with phone identity, we design two novel kernel functions that compute similarities between phone instances/types with same phone identities across pairs of utterances. With these kernels we train only a single SVM classifier for each pair of dialects.

We have seen that the kernel-based approach gives us the best results for recognizing the four Arabic dialects. When supervectors are constructed from means of GMMs of CD-HMMs, the kernel-based approach performs as well as the Discriminative Phonotactics approach (EER=6.0, EER=5.9%, respectively). The kernel-based approach is significantly improved when the supervectors are constructed from CI-phones modeled with GMMs (EER=4.9%). We have found that comparing supervectors of phone types as opposed to phone instances improves the EER (4.35%) and substantially improves the time complexity of the algorithm. Finally, using an upper bound of a KL-divergence-based kernel yields a significant reduction in EER (3.96%). Attempting to model features extracted from a larger context beyond monophones does not yield significant gains. Our kernel-based approach also outperforms a state-of-the-art approach (12.9% relative improvement of EER) on Arabic dialect recognition.

Chapter 10

Experiments for other Languages/Dialects

10.1 Introduction

Thus far, we have evaluated multiple approaches for dialect recognition on four broad Arabic dialects. We have found that Kernel-GMM-Type-KL is our best performing approach. In this chapter, we further evaluate the performance of this approach on Arabic sub-dialects to test whether our approach is able to capture more subtle differences between dialects. In addition, to test whether our approach generalizes for dialects and accents of other languages, we evaluate it: (1) on two NIST tasks: American English vs. Indian English accents and Southern vs. Non-Southern American English; (2) on American English at the state level plus Canadian English; (3) and on three Portuguese dialects.

10.2 Arabic Sub-dialect Recognition

10.2.1 Materials

We saw in Section 4.3 that each of the three Appen corpora (Levantine, Gulf, and Iraqi) includes multiple sub-dialects. Appen provides a "region" field for each speaker. Unfortunately, this field is not well documented. So, for our experiments, we assume that this field is the regional origin of the speaker. The Levantine corpus contains four regions for the four countries in Levantine: Jordan, Israel/Palestine, Lebanon, and Syria. Similarly, the Gulf corpus contains three countries: Oman, Saudi-Arabia, and UAE. However, the Iraqi corpus contains regions within a single country, Iraq: Baghdad, North, and South.

For each sub-dialect in these three corpora, 80% of the speaker files are used for training and 20% for testing. Recall that these Appen corpora contain male and female speakers speaking by landline or mobile phones. Similar to the construction of the test set of DATA II, for each sub-dialect, we randomly selected: 25% from the set of female speakers speaking on mobile phones; 25% from male speakers speaking on mobile phones; 25% from females speaking on landline phones; and 25% from males speaking over landlines. Unfortunately, this selection may lead to a small number of test speakers overall for dialects that lack sufficient speakers for any individual category. Therefore, we may have limited confidence in our conclusions about the difficulty of identifying these dialects. We test our approach here on 30s-long segments, so each speaker file is segmented to 30s cuts (after removing silence). We also use multiple cuts from the same speaker. See Table 10.1, for the number of speakers and cuts for each sub-dialect for training and testing.

Dialect	# Train Speakers	# Train Cuts	# Test Speakers	# Test Cuts
Baghdad	241	1546	60	400
North-Iraq	57	394	16	107
South-Iraq	84	548	20	154
Oman	112	620	28	156
Saudi-Arabia	471	3054	120	751
UAE	197	1154	48	271
Jordan	188	1076	48	292
Lebanon	198	1068	52	287
Israel/Palestine	204	1232	52	323
Syria	195	1215	48	285

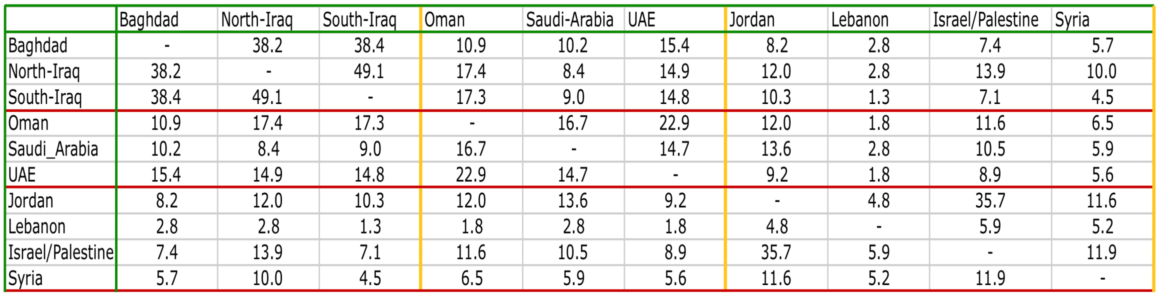
Table 10.1: Number of speakers and cuts for each sub-dialect for training and testing

10.2.2 Results

We test the Kernel-GMM-Type-KL approach on every pair of sub-dialects, as described above, and report the EER on the corresponding test set. Recall that, in this approach, we need to train an SVM binary classifier for each pair of dialects. We have ten dialects; thus we train 45 classifiers.

As shown in Table 10.1, some dialects contain far fewer training speakers than others. We observe that the SVM does not perform very well on imbalanced data for this task.[1] In this work, we have attempted to resolve the problem by giving different weights to each class, but unfortunately this does not improve the results. We find that balancing the training data by downsampling to the minority class performs better than weighting or than training on imbalanced data.

The EER for each pair of dialects is shown in Figure 10.1. The average EER of all pairs is 12.2%. Unsurprisingly, we find that the most confusion occurs between sub-dialects of the same broad dialect: the average EER between sub-dialects of the Gulf is 18.1%; the average EER between sub-dialects of Levantine is 15.8%; and the highest confusion occurs between sub-dialects of Iraqi (average EER is 41.9%). Note that we have a small number of training speakers for North and South of Iraq. This may have influenced the results; it is likely that, the more subtle differences between dialects, the more training data is required. Comparing the average EER across broad dialects, we obtain 5.4% for Iraq vs. Levantine, 5.6% for Gulf vs. Levantine, and 13.1% for Gulf vs. Iraq.



	Baghdad	North-Iraq	South-Iraq	Oman	Saudi-Arabia	UAE	Jordan	Lebanon	Israel/Palestine	Syria
Baghdad	-	38.2	38.4	10.9	10.2	15.4	8.2	2.8	7.4	5.7
North-Iraq	38.2	-	49.1	17.4	8.4	14.9	12.0	2.8	13.9	10.0
South-Iraq	38.4	49.1	-	17.3	9.0	14.8	10.3	1.3	7.1	4.5
Oman	10.9	17.4	17.3	-	16.7	22.9	12.0	1.8	11.6	6.5
Saudi_Arabia	10.2	8.4	9.0	16.7	-	14.7	13.6	2.8	10.5	5.9
UAE	15.4	14.9	14.8	22.9	14.7	-	9.2	1.8	8.9	5.6
Jordan	8.2	12.0	10.3	12.0	13.6	9.2	-	4.8	35.7	11.6
Lebanon	2.8	2.8	1.3	1.8	2.8	1.8	4.8	-	5.9	5.2
Israel/Palestine	7.4	13.9	7.1	11.6	10.5	8.9	35.7	5.9	-	11.9
Syria	5.7	10.0	4.5	6.5	5.9	5.6	11.6	5.2	11.9	-

Figure 10.1: EER for each pair of Arabic sub-dialects

[1] This is also a challenge for speaker verification; see [Mak and Rao, 2011] for a literature review of this problem and a new idea for resolving it for speaker verification.

10.2.3 Dialect-Map Visualization

In this section, we are interested in visualizing a two-dimensional map based on similarities between dialects using Multi-Dimensional Scaling (MDS). We would like to see the correspondence between this *dialect map* and the geographical map of the regions where the ten dialects are spoken. For MDS, we need a pairwise similarity matrix to describe the similarity between each pair of dialects. MDS assigns a location to each element (i.e., dialect) in the D-dimensional space; in this work $D = 2$.

One could suggest multiple ways to define distances between dialects. For example, symmetric KL-divergence between the distributions of lexical items (including morphemes), phones/phonemes, or phonotactics can be calculated as distance metrics across pairs of dialects. However, such metrics may require linguistic knowledge and analyses for each individual dialect. For example, we will need to develop shared orthographic and phonemic spelling systems as well as morphology analyzers for all dialects. In this work, we focus on phonetic differences based on our system's performance using acoustic data only.

Since our dialect recognition approach models phonetic features, it is reasonable to assume that, the higher the error (i.e., EER) between a pair of dialects using our system on a held out set, the more phonetically similar the pair is. In other words, the harder for the system to distinguish the pair of dialects, the more phonetically similar they are. We use the confusion matrix in Figure 10.1 as the input to MDS. After running MDS, we obtain the map in Figure 10.2.

We compare the resulting dialect map in Figure 10.2 to the geographical map in Figure 10.3. We observe that the locations of the dialects on the dialect map correspond closely to their locations on the geographical map for all dialects except for Oman and UAE. In the geographical map, Oman and UAE are located in the most southeastern sector. However, on the dialect map they are located south or southeast of every dialect except for Saudi-Arabia, which they are north of. This may suggest that these two dialects are more similar in terms of their phonetic structure to the rest of the dialects than to Saudi-Arabian. Although the general structure/orientation of the dialect map is still preserved, the Lebanese dialect is separated from the rest of the dialects. This suggests that Lebanese has a distinct phonetic structure. We also observe that Palestinian and Jordanian dialects are very close

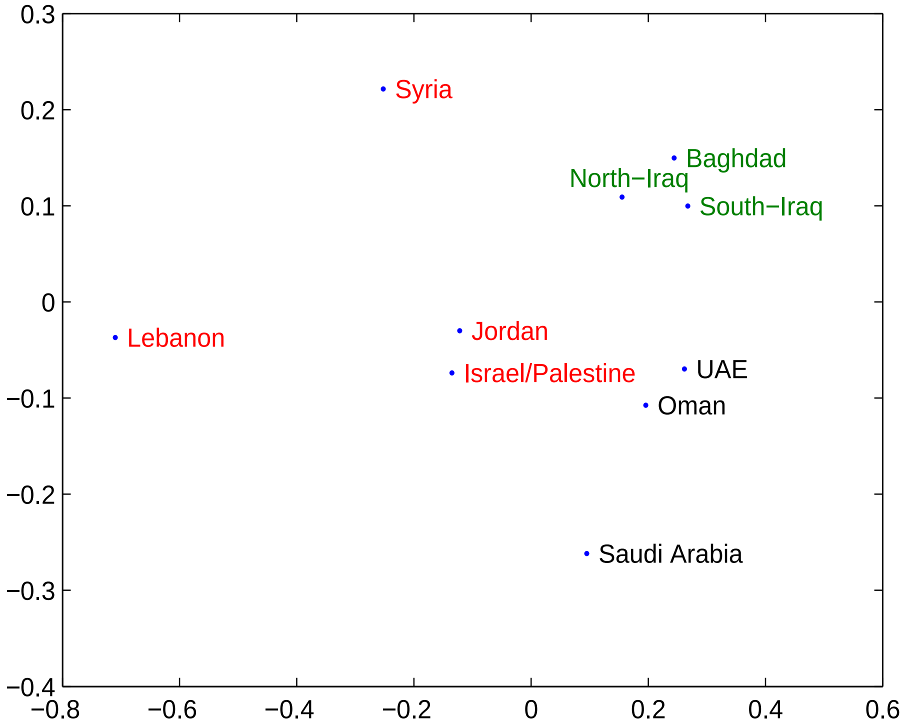


Figure 10.2: Dialect map using MDS

to each other on the dialect map. This can be explained by the fact that the majority of Jordanians are descended from Palestinians. Also, according to the United Nations Relief and Works Agency (UNRWA) for Palestine refugees in the near East, the Palestinian refugees in Jordan as of June 2008, were 31.5% of Jordan's population. We also find that Levantine dialects are the most scattered.

While this type of visualization may provide linguistic knowledge about the degree of differences between dialects, it can also aid, for example, in deciding what dialects should be grouped together when training an Arabic ASR system's acoustic models. For example, we might decide that the Lebanese dialect should have its own set of acoustic models, while the acoustic data for all the Iraqi dialects might be grouped together. Note that our system relies upon acoustic-phonetic features; thus the decision for the choice of language model would be a separate decision.

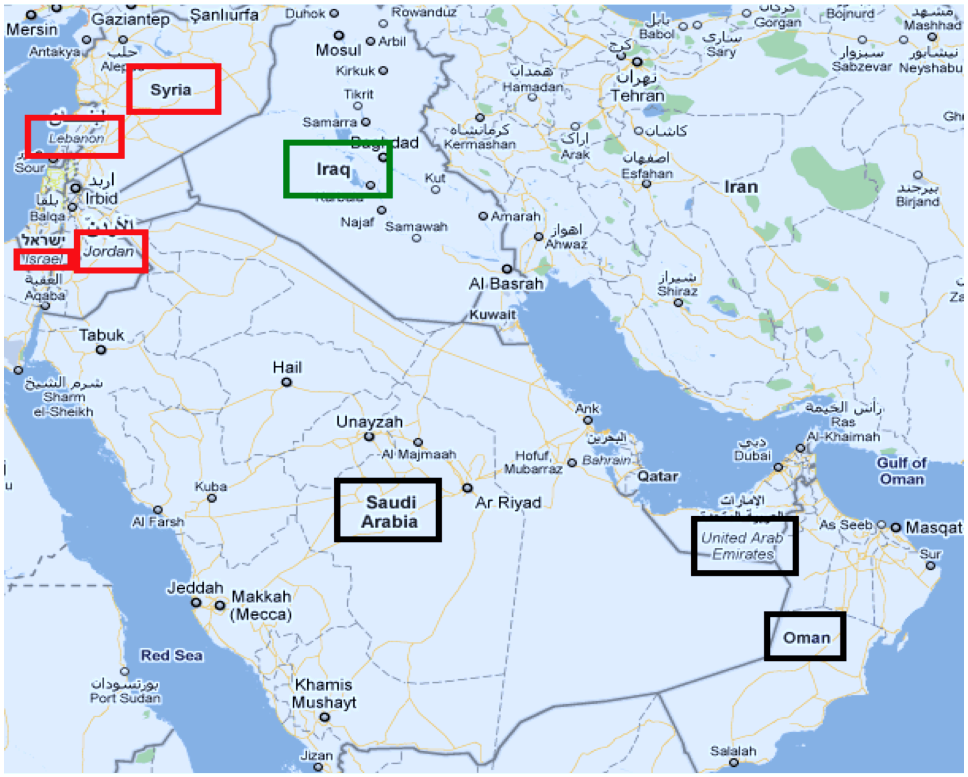


Figure 10.3: The geographical map of part of the Arab world

10.3 American vs. Indian English Accent Recognition (A NIST Task)

For the past few years, NIST has been introducing new dialects and accent tasks in the Language Recognition Evaluation (LRE). In this section, we test our system on one of the 2007 evaluation tasks: the recognition of American English vs. Indian English.

10.3.1 Materials

The American English training data are from the following telephone conversation corpora, totaling about 83 hours of speech: The American English speaker training files (30s cuts) form the 2005 NIST LRE [Le, 2005]; The CallHome American English Speech corpus [Canavan *et al.*, 1997a]; and 26.6 hours of randomly selected speech of Native American English speakers from Fisher English Training Part 1 [Cieri, 2005].

Our Indian English Accent training data are from the following telephone conversation corpora: The Indian English speaker training files (30s cuts) form the 2005 NIST LRE. The 14.5 hours of speech of Indian and Tamil English speakers from Fisher English Training Part 1 and 2. On the assumption that speakers carry aspects of the articulatory and coarticulatory effects from their native language to a non-native language, we augment our Indian accent training data with CallFriend Hindi Speech corpus [Canavan and Zipperlen, 1996d], following [Torres-Carrasquillo *et al.*, 2008]. Note that we can make use of such data particularly because our system models low-level phonetic features only. The total duration of our Indian training data is about 40 hours of speech.

We segment both corpora to 30s training segments and use multiple segments from the same speaker. We end up with 2589 Indian training segments, and 4877 American English training cuts.

10.3.2 Evaluation

We test our kernel-based system on the official 2007 NIST LRE Test Set (the 30s task) [Martin and Le, 2007]. This set contains 79 American English speakers and 160 trials of Indian English speakers. This official set allows to directly compare the performance of our approach to published work and to the winning system in NIST LRE 2007.

10.3.2.1 Baselines

We saw in Section 9.6 that Kernel-GMM-Type-KL outperforms Torres-Carrasquillo et al. [2008]'s system on the three broad Arabic dialect recognition task. Torres-Carrasquillo et al. [2008] also test their state-of-the-art approach on the American vs. Indian English accent task, employing a superset of the training data described in Section 10.3.1. They obtain an EER of 10.6%. The EER using the GMM-SVM approach [Campbell *et al.*, 2006b] on this task, according to [Torres-Carrasquillo *et al.*, 2008] is 11.3%. Chen et al. [Chen *et al.*, 2010] evaluated their system on this task as well. Their system identifies dialect-discriminating phonetic rules which are subsequently used to adapt biphone models. These models are then used for scoring a test trial for recognition (see Section 4.2 for more details). Their approach achieves an EER of 14.7%, and, when fused with the PRLM approach, they obtain an EER of 10.6%, similar to [Torres-Carrasquillo *et al.*, 2008]. We compare the performance of our approach to these systems.

10.3.2.2 Results

To train the Kernel-GMM-Type-KL system on the English corpora, we need English phone hypotheses. We employ the English phone recognizer from the Brno University of Technology [Matejka *et al.*, 2005]. The phone inventory (Φ) of this recognizer consists of 38 phones (phone types). For each phone type, we ML-train a phone-GMM of 60 Gaussian components using a random sample of frames from the training data, aligned to phone instances of this phone type. For this task, the acoustic features are 13 RASTA-PLP features (including energy) plus delta and delta-delta, resulting 39D feature vector from each frame.[2] The rest of the steps/settings of this system are exactly the same as those of the Kernel-GMM-Type-KL of the Arabic experiments.

We train another system which is based on the Kernel-GMM approach. Recall that, in this approach, we compare phone instances between utterances as opposed to phone types. Instead of the RBF kernel in (9.1), the kernel function which corresponds to the approximation of the GMM-KL-divergence in (9.11) is utilized for this system. We denote this system Kernel-GMM-Instance-KL.

As shown in Figure 10.4, evaluating these two systems on the the official 2007 NIST LRE test set, we obtain an EER of 8.7% for Kernel-GMM-Type-KL and slightly, but not significantly, better ERR of 7.8% using Kernel-GMM-Instance-KL. Combining the output of these two systems by simply averaging the SVM posteriors, we achieve the best EER: 6.3%. All these systems outperform both [Torres-Carrasquillo *et al.*, 2008] and [Chen *et al.*, 2010]'s systems. Although the combination system achieves 40.6% relative improvement in EER, the improvement over both baselines is not statistically significant, due to the small number of test trials. Also, as shown in Figure 10.5, our combination system performs better than the winning 2009 NIST system on this data set.

[2] We use the Audio feature calculation program *feacalc* version 0.91 from ICSI's SPRACHcore software package.

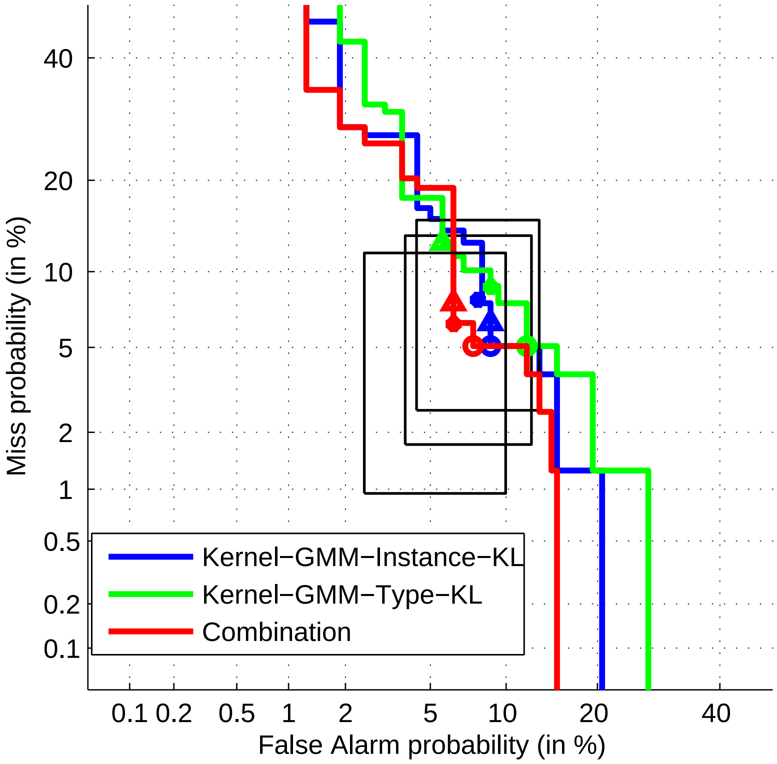


Figure 10.4: Comparing the DET curves of our approaches on the 2007 NIST LRE test set for American vs. Indian English

10.4 American English Dialects: Southern vs. Non-Southern (A NIST Task)

In this section, we evaluate our Kernel-GMM-Type-KL system on the 1996 NIST LRE dialect task: Southern American English vs. Non-Southern. We compare the performance of our system to the GMM-UBM approach.

10.4.1 Materials

The speakers in the "southern" collection are from the CallFriend American English-Southern Dialect corpus [Canavan and Zipperlen, 1996b]. The speakers were identified primarily based on vowel quality patterns common among native speakers raised in the southeastern

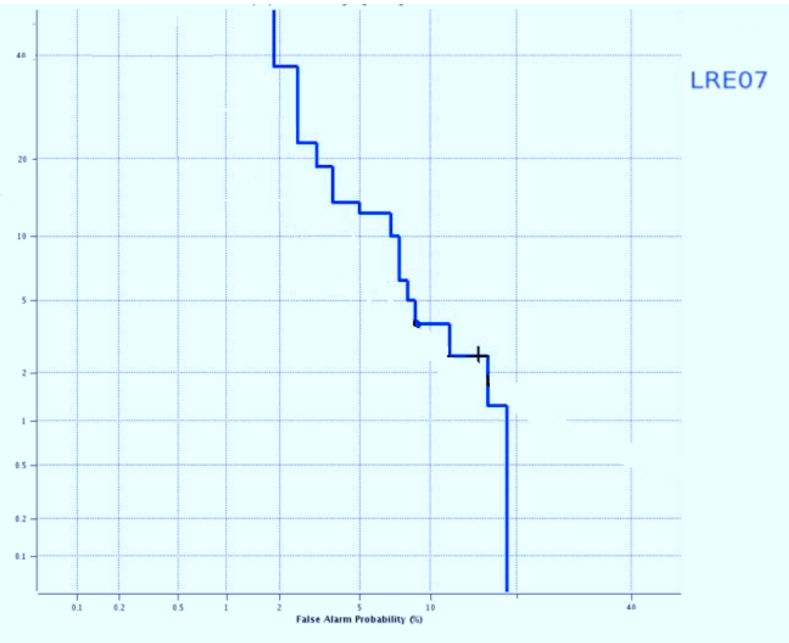


Figure 10.5: The DET curve of the best performing 2007 NIST system on the American vs. Indian English task (screenshot from the 2009 NIST workshop slides)

United States (from Texas eastward to the Atlantic coast, and from Virginia and Kentucky southward to the Gulf of Mexico). This corpus also includes a small number of African-American speakers, whose geographic origins may be more dispersed, but who share some of the vowel quality patterns distinctive of southern white speakers. The dialects of these speakers were verified by a human subject who is a native speaker familiar with American English dialects.

This corpus is divided into 40 speakers for training, 40 for development, and 40 for testing. In this work, we use both the training and development portions to train our models and the 40 test speakers to evaluate the system. Similar to our other experiments, we initially segment the speech files based on silence, and then extract multiple cuts of about 30s each from each speaker's data (see Table 10.2).

The speakers in the "non-southern" collection are from the CallFriend American English-Non-Southern Dialect corpus [Canavan and Zipperlen, 1996a]. The speakers in this corpus are from a wide geographic range, based on their own reports of where they were raised. Some of these speakers identified their origins as being in the southeastern U.S. However,

Dialect	# Train Speakers	# Train Cuts	# Test Speakers	# Test Cuts
Southern	80	1694	40	839
Non-Southern	80	1816	40	871

Table 10.2: Number of speakers and cuts for each American dialect

according to the LDC, regardless of the speakers' geographic or ethnic backgrounds, all of them share clear absence of a vowel quality pattern that would distinguish them as speakers of a "southern" dialect.

This corpus contains as many speakers as in the southern corpus, and also is divided into 40 speakers for training, 40 for development, and 40 for testing. In this work, we also use both the training and development portions for training and evaluate on the 40 test speakers. As shown in Table 10.2, we again segment the files based on silence, and then segment each speaker's audio data to multiple cuts of about 30s each. It is important to note that, in both corpora, the participants on both sides of the telephone conversations spoke in the same dialect. This avoided possible phonetic entrainment between the speakers on both sides.

10.4.2 Evaluation

As far as we can determine, there is no published work which reports individual results (DET curves or EERs) for this task. Also, this task is not in any recent NIST LRE, and no results are presented in any NIST Speaker and Language Recognition workshop. Therefore, we choose the standard GMM-UBM approach as our baseline (see Chapter 7 for the GMM-UBM approach).

Similar to the American vs. Indian English accent experiments, for the GMM-UBM, we extract 39D RASTA-PLP acoustic features. We use the training files in both corpora to ML-train the GMM-UBM, of 2048 Gaussians. For each dialect, using the training cuts of this dialect, we build a GMM by MAP adapting the UBM with 5 iterations using relevance factor $r = 16$. We score the test cuts in each dialect using PairScoring scheme to plot the DET curve. As shown in Figure 10.6, the EER is 31.4%, significantly above chance.

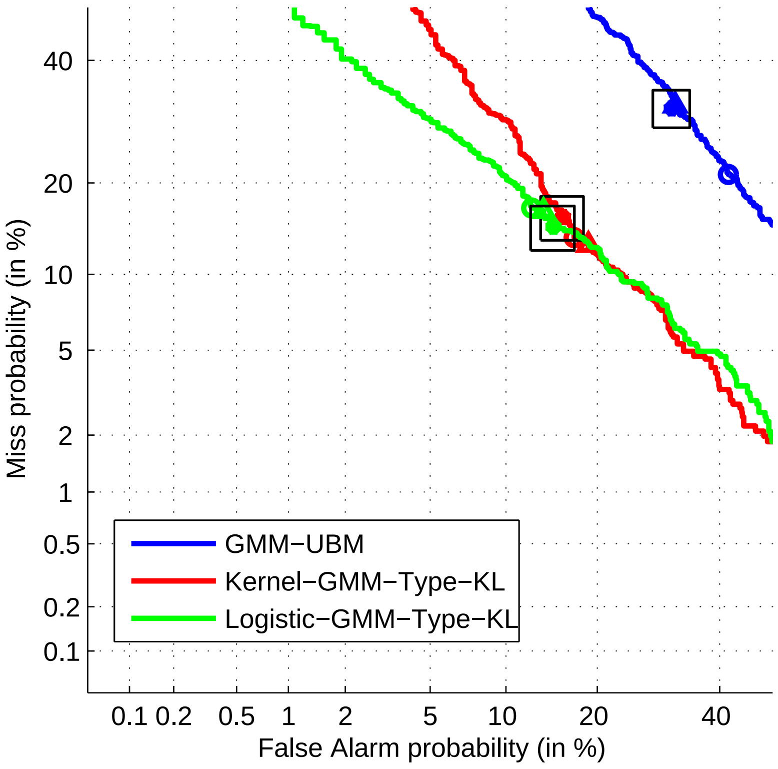


Figure 10.6: The DET curves GMM-UBM, Kernel-GMM-Type-KL and Logistic-GMM-Type-KL for American English Southern vs. Non-Southern dialects.

Now we compare the GMM-UBM performance to that of the Kernel-GMM-Type-KL system on these test cuts. The Kernel-GMM-Type-KL system here has the same settings as in the American vs. Indian English experiments, but is trained on the corpora described above. We obtain an EER of 15.7% (absolute), a 53.5% relative improvement over the GMM-UBM, a significant improvement (see Figure 10.6). Recall that, for this task, we have only 80 training speakers per dialect, a relatively small number comparing to our Arabic and American vs. Indian English experiments. Form our experience on Arabic, we believe that including more training speakers, our system will achieve better results.

We mentioned in Section 9.4.4.2 that, for the Kernel-GMM-Type-KL approach, an utterance can be represented as a single vector W of supervectors.[3] Such a representation

[3] The supervectors are scaled and shifted using the weights, inverse covariance matrices and means of the

allows us to experiment with a different classifier than SVM. We experiment with logistic regression using this vector representation for the Southern vs. Non-Southern American English task. We train a logistic regression with L_2 regularizer on the same vectors used for Kenel-GMM-Type-KL above and test on the same test cuts. Unsurprisingly, as shown in Figure 10.6, due to the close relationship between SVM (with linear kernel) and logistic regression, the logistic regression classifier on this task performs slightly but not significantly better (EER of 14.6%) than the SVM using the kernel in (9.12) [Vapnik, 1999]. We denote this approach Logistic-GMM-Type-KL. Note however that the DET curve corresponding to the logistic regression in Figure 10.6 has a slope closer to -1 than the SVM DET curve. Hence, the logistic regression posteriors (confidence scores used to plot the DET curves) were drawn from more *standard* normal distribution.[4]

10.5 American English Dialects: Pairwise State Classification

We saw in the previous section that Southern and Non-Southern American English regional dialects can be distinguished from one another with considerable accuracy. In this section, we are interested in identifying dialects at the American state level. In other words, we identify those pairs of American states whose dialects are distinguished from one another. While it is known that some of these states do not have their own distinct dialects, we would still like to use our system to help address one of Labov's questions: "What are the major dialect regions of the United States?" [Labov *et al.*, 1997]. As in our Arabic subdialect experiments, we visualize an American state dialect map based on pairwise dialect similarities using EERs obtained from our system.

10.5.1 Materials

For the experiments in this section, we employ the Fisher English Training Part 1 and 2 corpora [Cieri, 2005]. These corpora consist of conversational telephone speech, created

corresponding Gaussians of each phone-GMM (see Section 9.4.4.2).

[4] Confidence scores normally distributed with a variance of 1 are crucial for speaker verification.

at the LDC during 2003. Each audio file contains a full English conversation of up to 10 minutes. In this work, we use only audio the files whose speakers are native North American English speakers. In addition to the annotation of whether a speaker is a native English speaker or not, speakers are annotated with the North American state where they were raised. In our experiments we also include native English speakers from Canada as a single separate class.[5]

For each state, we randomly select 80% of the speakers for training and hold out the rest (20%) for testing. To avoid cases were we have insufficient data, we limit our experiments to states that have at least 50 training speakers. With such constraints, we end up with 31 American states plus Canada. Similar to our other experiments, we test our system on 30s cuts (after removing silence), with multiple cuts from the same speaker. See Table 10.3 for the number of the training and testing speakers and cuts for each American state.

It is important to note that these corpora are not designed for the task of dialect recognition. There are no annotations about the number of years the speakers lived in their native states. Speakers may have moved to different parts of the country, so their native dialects may have faded. Moreover, there is no dialect-based assessment, as for the NIST Southern vs. Non-Southern corpus, in Section 10.4. Due to the lack of such knowledge, we cannot judge the real difficulty of the task from our experiments.

10.5.2 Results

Using the Kernel-GMM-Type-KL approach with the same settings as in the previous English experiments (Sections 10.3 and 10.4), we train a binary SVM classifier for each pair of states on the corresponding training cuts in Table 10.3. We balance the training data by downsampling to the minority class. For the 32 states, we therefore train 496 classifiers and evaluate them on the corresponding held-out test cuts. We report the EER for each pair of classifiers in Tables A.1–A.7.

To summarize our classification results, we categorize the American states using Labov et al. [1997]'s dialect categorization. We then compute the average EER within each category and the average EER across classes. We report these numbers in Tables 10.4 and Table 10.5;

[5] There are no state-level annotations for Canadian speakers.

Dialect	# Train Speakers	# Train Cuts	# Test Speakers	# Test Cuts
AL	159	873	21	115
AZ	143	798	19	109
Canada	200	1141	24	134
CA	1914	10639	235	1338
CO	142	807	21	121
CT	278	1552	33	190
FL	538	3038	69	387
GA	382	2156	53	297
IA	165	915	21	122
IL	761	4288	94	535
IN	321	1796	42	249
KS	140	789	18	102
KY	169	942	21	122
LA	152	868	21	123
MA	449	2531	58	324
MD	286	1630	39	227
MI	586	3313	69	381
MN	314	1776	38	215
MO	244	1387	31	175
NC	256	1473	34	197
NJ	807	4582	99	567
NY	2295	13125	276	1586
OH	681	3829	88	497
OK	189	1104	23	128
OR	187	1053	25	134
PA	1649	9391	195	1105
SC	111	620	16	93
TN	255	1436	29	169
TX	815	4587	107	595
VA	296	1643	39	221
WA	359	2008	44	243
WI	243	1360	33	180

Table 10.3: Number of speakers and cuts for each American State

recall that chance EER is 50%.

We observe that Canadian English is the most distinguished from all other American dialect categories. Also, Canadian English is most similar to the Eastern dialects. We also find that the Southern and the Western dialects are the easiest to distinguish from one

Dialect Pair	Average EER (%)	Sdev
Canada vs. The South	20.2	5.12
Canada vs. The Midland	23.89	5.6
Canada vs. The West	27.76	3.73
Canada vs. The East	29.34	2.87
The South vs. The West	30.47	7.12
The South vs. The East	32.8	6.33
The East vs. The West	34.78	3.9
The Midland vs. The West	35.5	6.09
The Midland vs. Th South	36.27	7.6
The Midland vs. The East	36.31	6.26

Table 10.4: Average EER across dialect categories

Dialect Pair	Average EER (%)	Sdev
The West	36.65	6.73
The East	37.25	5.58
The Midland	37.91	6.93
The South	43.93	8.03

Table 10.5: Average EER within dialect category

another (EER of 30.47%). Note that this error is far greater than the error we obtain for the Southern vs. Non-Southern experiment in the previous section (EER of 15.7%). We speculate that this difference may be due to the major differences between the two corpora. As discussed in Section 10.5.1, unlike the Fisher corpora, the Southern vs. Non-Southern corpus is designed specifically for dialect recognition. Moreover, the dialects of all speakers were manually verified by a linguist. In addition, the participants on both sides in this corpus spoke the same dialect – to avoid dialect entrainment. From Table 10.5, we observe that the dialects of the Southern states are the most similar to each other, whereas the dialects of the Western states are the most diffused.

As in our Arabic experiments, we are interested in visualizing our results in a two-dimensional dialect map using MDS. To construct the pairwise similarity matrix for MDS, we utilize the EER of each pair of states as the dialectal similarity measure between this pair. The output map is shown in Figure 10.7. We can see that this map resembles the actual geographical map somewhat. If we impose manual clustering somewhat similar to Labov's work, we obtain Figure 10.8. We can see in this map that only California, Colorado, Iowa, and North Carolina are geographically misplaced. The Californian dialect is located on the middle of the map. This may be due to the substantial migration to California from different parts of the country. We believe that this empirical dialect map may be of an interest of dialectologists and speech scientists as well as engineers who, for example, would like to build dialect-specific ASR acoustic models.

10.6 Portuguese Dialect Recognition

In this section, we are interested in testing whether the Kernel-GMM-Type-KL approach generalizes for dialects of languages other than Arabic and English. So, we evaluate the Kernel-GMM-Type-KL on three Portuguese dialects: African Portuguese (AP), Brazilian Portuguese (BP), and European Portuguese (EP).

10.6.1 Materials

We employ the same training and testing data sets and segmentations used in [Koller, 2010; Koller *et al.*, 2010] (see Tables 10.6 and 10.7). The AP data set is a collection of broadcast news and soap operas from RTP-Africa, a terrestrial television channel for the Portuguese-speaking African countries. It has been pointed out by Koller et al. [2010] that many of the African speakers, namely reporters and politicians, were educated in Portugal. According to human subjects in a perceptual study, these African Portuguese reporters and politicians are hardly distinguished from European Portuguese speakers. Also, some of the African speakers are not native speakers of Portuguese. The BP data set contains recordings of several broadcast news and conversations. The EP data set consists of a selection of about eight hours from the ALERT Speech Recognition Training corpus. These

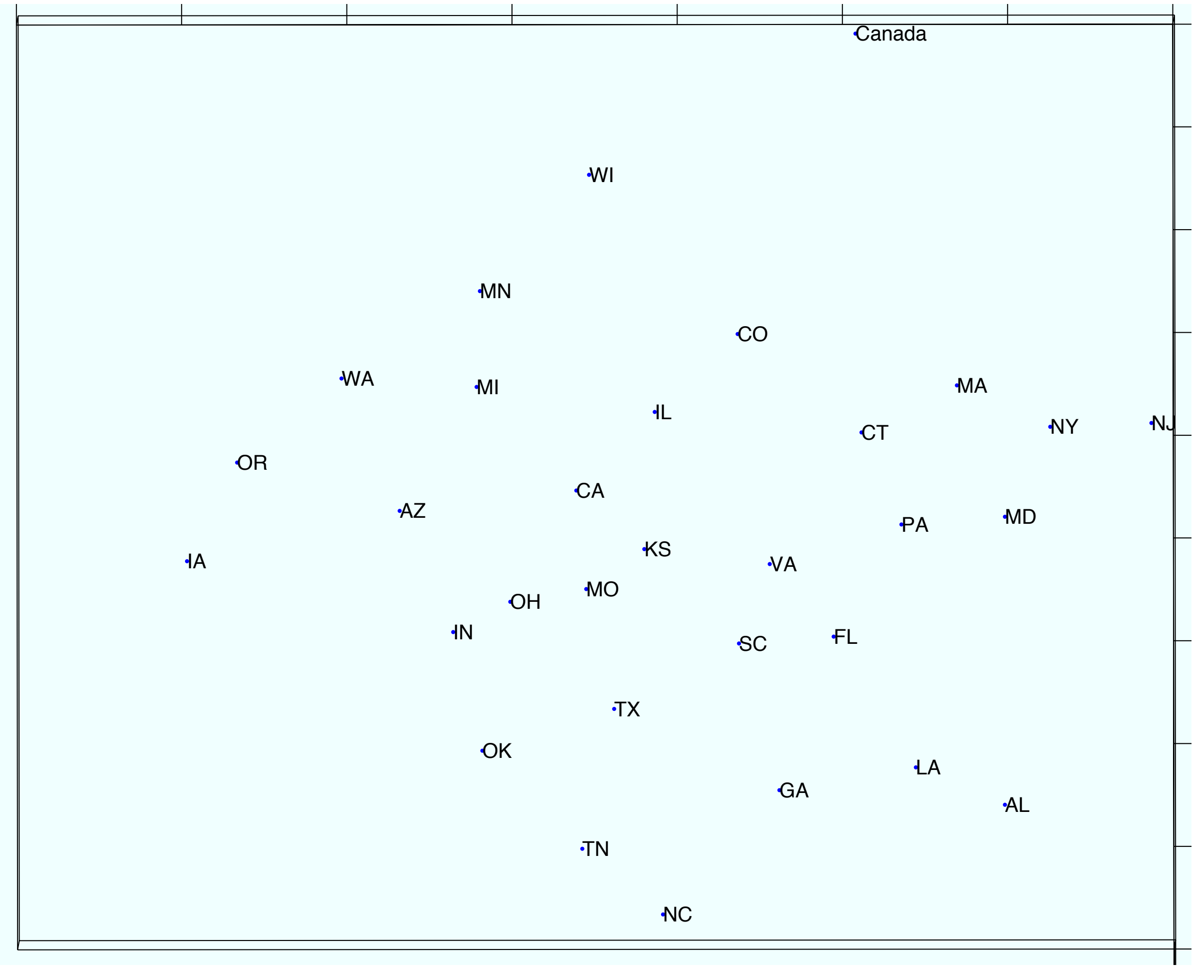


Figure 10.7: American State map using MDS

corpora are described in greater details in [Koller, 2010].

It should be noted that, unlike all our previous experiments, the data here consists mostly of broadcast news shows (sampled at 16Khz) as opposed to telephone conversations, which is sampled at 8Khz. Another difference from our other experiments is that most of the segments here are substantially shorter than 30s (see Tables 10.6 and 10.7).

10.6.2 Evaluation

Using the Kernel-GMM-Type-KL approach, we train three binary SVM classifiers, one for each pair of the three Portuguese dialects. To build such systems, we need Portuguese phone hypotheses for each segment. In this work, we run the European Portuguese phone

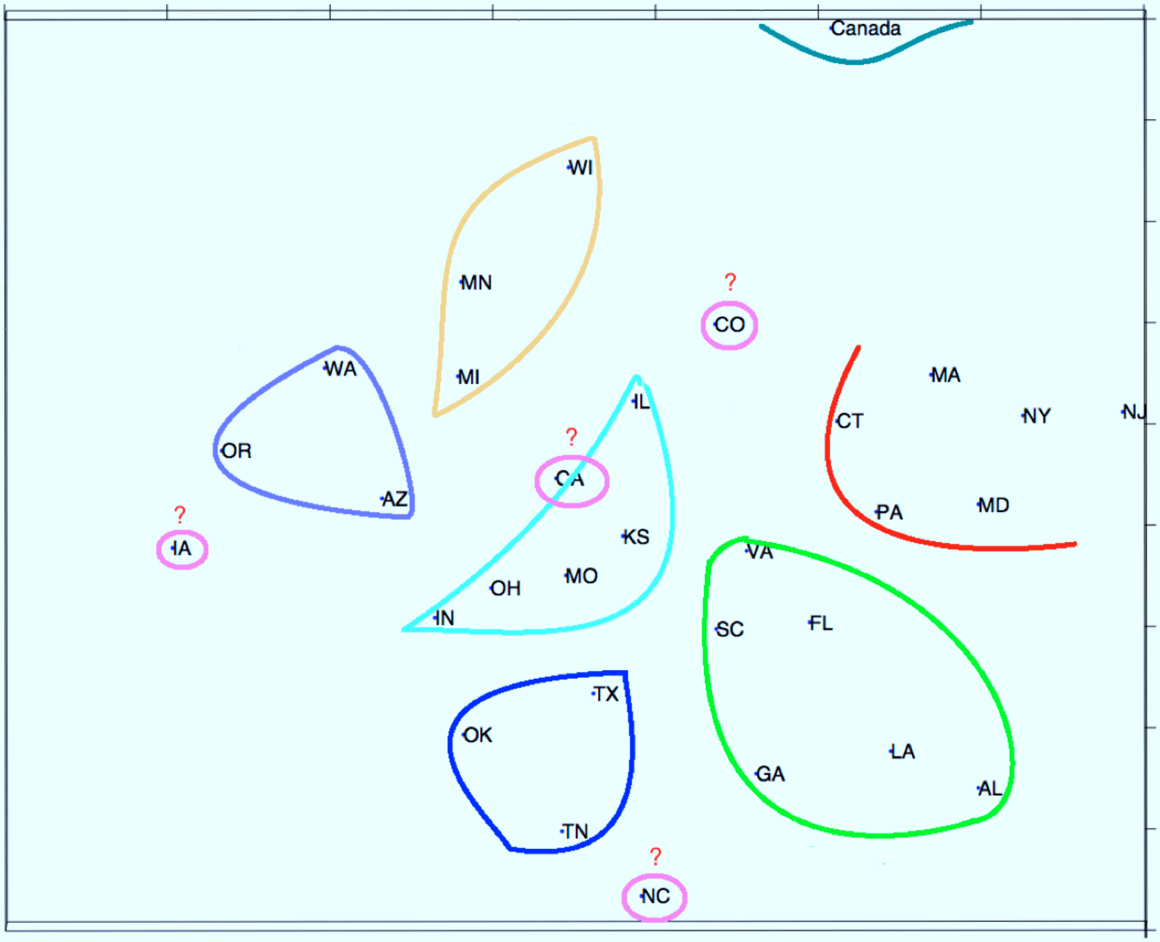


Figure 10.8: American State map using MDS with manual clustering somewhat similar to Labov's work

recognizer developed by Koller et al. [2010]. The phone inventory of this phone recognizer consists of 38 phones. The acoustic features, aligned to each phone instance, are also 39D vectors: 13 RASTA-PLP features (including energy), delta and delta-delta.[6] Similar to the American vs. Indian English experiment, the phone-GMM are of 60 Gaussian components and are trained for each of the 38 phone types using all the training data from the three dialects.

Figure 10.9 shows the results of our system on the test segments (in Table 10.7) for

[6] We thank Alberto Abad Gareta for running the phone recognizer and extracting these PLP features for all the Portuguese data.

Training Data	AP	BP	EP
Duration (min)	238.8	256.1	279.1
# Segments	1424	1434	1283
% Segments ($< 3s$)	10.1	10.7	13.1
% Segments ($3 - 10s$)	42.3	44.4	49.6
% Segments ($10 - 30s$)	38.7	38.7	44.1
% Segments ($> 30s$)	2.2	4.1	6.3

Table 10.6: Portuguese Training Data

Testing Data	AP	BP	EP
Duration (min)	88.8	80.2	99.0
# Segments	610	462	412
% Segments ($< 3s$)	23.3	18.0	0.2
% Segments ($3 - 10s$)	43.1	40.0	42.5
% Segments ($10 - 30s$)	32.8	38.1	50.0
% Segments ($> 30s$)	1.0	4.1	7.5

Table 10.7: Portuguese Testing Data

each of the pairs of Portuguese dialects. We observe that the most distinguishable pair of dialects is BP vs. EP (EER=8.0%). This is not surprising due to the major phonetic differences between this pair of dialects, including vowel reduction [Rouas *et al.*, 2008; Abad *et al.*, 2009]. We obtain an EER=9.0% for AP vs. BP. We find that the most difficult pair to distinguish is AP vs. EP (EER=11.9%). Nonetheless, none of these differences is statistically significant. Obtaining the highest error between AP vs. EP may be due to the similarities between the dialects/accents of the AP and EP speakers in these corpora (see Section 10.6.1). However, due to the unbalanced distribution of test segment durations across dialects, we cannot draw clear conclusions about the degree of similarities between these dialects.

It has been shown that some of these dialects differ in terms of their rhythmic structures

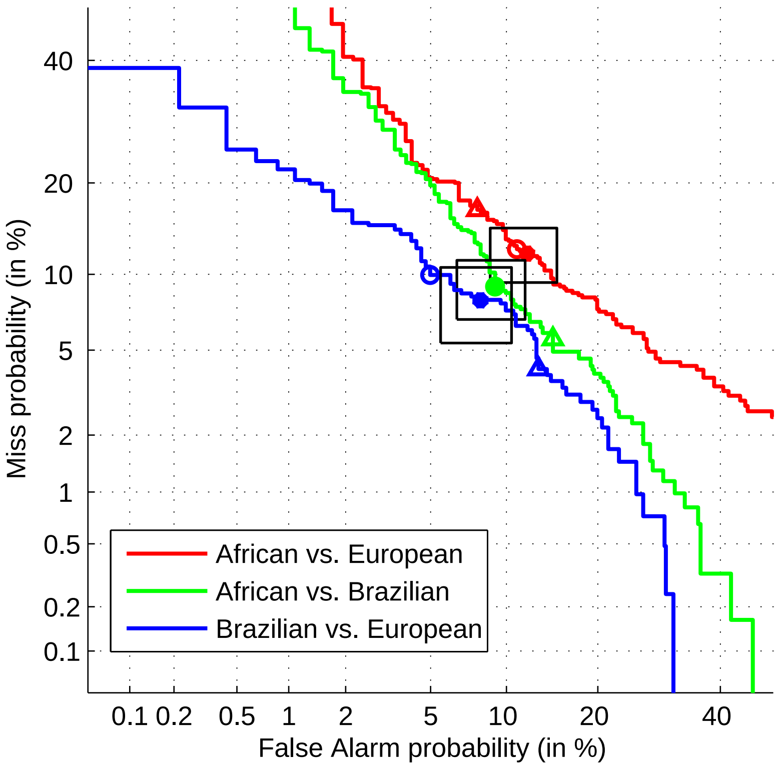


Figure 10.9: DET curve for each of the three Portuguese dialects (without rhythmic features)

[Frota and Vigrio, 2001]. We saw in Section 9.4.4.2 that, for the Kernel-GMM-Type-KL approach, each utterance can be represented in one vector W. This representation allows us to add new features easily. We experiment with the following durational and rhythmic features to W.

- The mean and standard deviation of the durations of phone instances of each phone type in the utterance – two features for each phone type (similar to our Arabic prosodic modeling experiments, Chapter 6);
- The mean and standard deviation of the durations of all vowels in the utterance;
- The percentage of vocalic intervals ($\%V$);
- The standard deviation of intervocalic intervals (ΔC).

Adding these rhythmic features, we obtain reduction in EER for distinguishing EP form the other two dialects: AP vs. EP (from 11.9% to 8.5%) and BP vs. EP (from 8.0% to 6.3%). However, for some reason these rhythmic features increase the EER for AP vs. BP (from 9.0% to 12.3%). See Figure 10.10 for the comparison between the DET curves of each pair of classifiers with and without the rhythmic features.

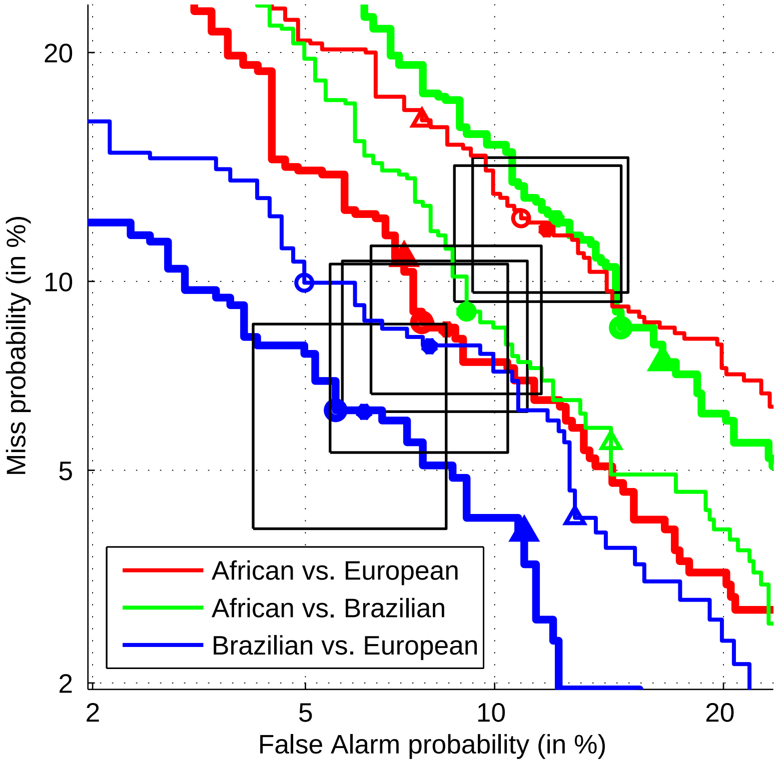


Figure 10.10: DET curve for each of the three Portuguese dialects (thick lines are with rhythmic features)

Now we compare our best results to those obtained by Koller [2010] on the same corpora and segments.[7] It is important to note that a main difference between our system and Koller's is that his system requires orthographic transcripts (during training) for each dialect to be identified. For our system we need only one phone recognizer, ideally, but not crucially

[7] Koller's system is described in detail in Section 4.2.

trained on one of the dialects.[8] In fact, we hypothesize that our system can work reasonably well using a phone recognizer trained on a different language. As shown in Table 10.8, Koller's system outperforms our approach for identifying the three pairs of Portuguese dialects, Nonetheless, none of theses differences is statistically significant. We speculate that the greater error of our approach over Koller's system may be due to the very short utterances in this data set. Recall that we need to MAP adapt multiple phone GMMs from the input utterance. With short utterances, these GMMs will not be robustly estimated, especially with errors introduced by the phone recognizer.

Dialect Pair	Koller's System (EER)	Kernel-GMM-Type-KL (EER)
AP vs. EP	4.1%	8.5%
EP vs. BP	5.9%	6.3%
AP vs. BP	7.6%	9.0%

Table 10.8: Comparing the EER of Koller [2010]'s System to the EER of our system on the three Portuguese dialects

10.7 Conclusions

In this chapter, we have shown that, not only does Kernel-GMM-Type-KL exceeds the state-of-the-art results for four broad Arabic dialects, but it is also effective for distinguishing most pairs of Arabic sub-dialects, and it outperforms state-of-the-art approaches on recognizing American English vs. Indian English Accents. Moreover, we see that this approach outperforms the standard GMM-UBM approach for Southern vs. Non-Southern American English, with 53.5% relative improvement. Our system can also be used to distinguish a number of dialects across American states plus Canada with considerable accuracy. In addition to Arabic and English, the system achieves comparable results to the state-of-the-art Portuguese dialect recognition system, a system which, unlike ours, requires orthographic transcripts during training. These experiments lead us to conclude that the Kernel-GMM-

[8] For example, in all our colloquial Arabic experiments we use a phone recognizer trained on MSA.

Type-KL is general enough for languages other than Arabic. Finally, we plot data-driven dialect maps using the pairwise EER between pairs of dialects for the Arabic dialects as well as for those of American states.

Chapter 11

Improving ASR using Dialect ID

11.1 Introduction

As discussed in previous chapters, one of the key challenges in Arabic speech recognition research is how to handle the differences between Arabic dialects. In this chapter, we suggest a method that utilizes our dialect recognition system in aid of improving ASR. Since Arabic dialects differ in their lexical items, we hypothesize that optimizing the language model to a specific dialect may improve Arabic ASR on this dialect. Similarly, since some phones and allophones are realized differently across Arabic dialects, training dialect-specific acoustic models may also improve performance. To build and utilize such dialect-specific models we have to annotate the data prior to training and recognition. To do that, we make use of our Arabic dialect recognition system.

Note that building a completely dialect-specific ASR system requires a third component to be specialized (or adapted) along with the language model and acoustic models: the pronunciation dictionary. In this chapter, we do not attempt to improve pronunciation modeling for Arabic colloquial dialects due to the lack of a morphological analyzer and a disambiguation tool, such as MADA (see Chapter 3), for Arabic colloquial dialects.

11.2 Dialect Identification System

Recall that our best performing dialect recognition system makes use of the kernel-based approach (Kernel-GMM-Type-KL). In this system, we train a binary SVM classifier for each pair of dialects, hence six binary classifiers for the four broad Arabic dialects (Levantine, Iraqi, Gulf, and Egyptian). To annotate the data with dialect ID tags, we need a single four-way classifier to classify the dialect of the speaker to one of the four dialects. To build such a classifier, we first run the six SVM binary classifiers on the held-out test set in Data II (4008 30s samples). Every SVM binary classifier provides a posterior probability $P(C_1|x)$ for each test sample x of belonging to class C_1. We use these posteriors as features to train a four-way logistic regression on the test cuts. The 10-fold cross validation of this classifier is 93.33%; the F-measure of each dialect class is shown in Table 11.2.

Dialect	F-Measure (%)
Levantine	90.7
Iraqi	86.7
Gulf	87.1
Egyptian	98.6

Table 11.1: F-Measure for each dialect class using the four-way classifier with 30s cuts

11.3 Data Sets

In this section, we describe three data sets we construct for our ASR experiments. All these sets are selected from the GALE project data collection released by the LDC. The data consists of broadcast news and conversations sampled at 16Khz. Recall that our dialect ID system is trained on telephone conversations; therefore, downsampling to 8Khz is required when running the dialect ID system on this data.

11.3.1 Baseline Data Set (BaseSet)

We use approximately 301 hours randomly selected from the entire GALE data collection (broadcast news and conversations) to train our baseline acoustic models. This data set contains 9,774 speakers. We denote this data set as BaseSet. Since the GALE collection consists mostly of MSA data, this set is likely to be composed of mostly MSA.

11.3.2 Hypothesized Dialect Data Set (DialectData)

In Broadcast Conversations (BC), the genre we attempt to improve the ASR system on, Arabic speakers often code switch between MSA and their native regional dialects. Code switching may occur in the multiple dimensions of the linguistic spectrum, including phonology, morphology, lexicon, and even syntax. For example, Arabic sentences may contain both MSA and dialectal phrases as well as MSA words with dialectal affixations.

The mixing of the two variants in the same utterance makes it difficult to select "clean" dialectal data to train and test our dialect-specific models. But, fortunately, some of the GALE broadcast conversations were manually annotated with MSA and non-MSA tags at the word level, by LDC.[1] So, in this section, we make use of these annotations to construct our training and testing data sets for each dialect as follows. We do that by selecting all utterances with more than 50% of the words annotated as non-MSA. To avoid the problem of having too little data for speaker compensation transforms, we retain speakers who have at least two utterances of at least 10 words. As a result, we obtain about 46.1 hours of speech from 1095 different speakers. We denote this data set as DialectData.

We run our four-way dialect classification system, described in the previous section, at the speaker level of DialectData to obtain dialect tags. The number of speakers and duration for each hypothesized dialect set is shown in Table 11.2. Since we lack true dialect annotations, we have manually evaluated a random sample of the hypothesized dialect tags and have found that the annotations are reasonable. Note that most of the speakers were classified as Levantine. This is not surprising since most of the data were collected from Levantine TV channels (such as, LBC, Alurdunya and SyriaTV).

[1] On the other hand, manual inspection of a sample of these annotations showed various deficiencies.

Dialect	# Speakers	Duration (hours)
Levantine	771	33.4
Iraqi	109	2.0
Gulf	74	1.8
Egyptian	141	9.0

Table 11.2: The distribution of hypothesized dialect data

Due to the biased distribution towards Levantine, in this chapter we focus our attention on improving the WER only on Levantine – i.e., we will optimize the language model for Levantine and build Levantine acoustic models. Nonetheless, we still test the Levantine-specific system on the other three dialects to test whether we obtain the most gain on Levantine. From the 33.4 hours of hypothesized Levantine, we construct three sets of 27.3, 3.0, and 3.1 hours for training (LevTrain), development (LevDev), and test (LevTest), respectively. We have no speaker overlap in these three sets. We use 3.0 hours from the 9.0 hour-hypothesized Egyptian data for testing our system on Egyptian. We select all the 2.0 and 1.8 hours of hypothesized Iraqi and Gulf, respectively, as our test sets for these two dialects. The total duration of the hypothesized dialectal data is about 9.9 hours; we denote this set as DialectTestSet.

11.3.3 Hypothesized Levantine BC (LevBC)

We have only 33.4 hours of speech annotated as non-MSA and automatically tagged as Levantine. This amount of data is not sufficient to robustly train an ASR system's acoustic models. Therefore, we run our dialect ID system on the entire GALE BC data regardless of whether the utterances are tagged as MSA or not. After running the dialect ID system, we obtain about 313 hours of speech annotated as Levantine. Note that this data set includes LevTrain but does not include LevDev or LevTest. It should also be noted that this set may contain noisy Levantine data, since some utterances may be completely MSA. We denote this data set as LevBC. The total number of speakers in this set is 5,291 speakers.

11.4 ASR System

We use IBM's Attila ASR system to conduct our experiments. Below we describe the design of the system used in all these experiments. The system design is similar to that described in [Saon *et al.*, 2010].

11.4.1 Front-End and Acoustic Models

The input speech is represented by 13-dimensional PLP features extracted from 25ms frames, with a frame-shift of 10ms, with cepstral mean and variance normalization. Each frame is spliced together with four preceding and four succeeding frames and LDA is performed to yield 40-dimensional feature vectors. The LDA matrix used here is from a previous IBM's ASR Arabic system [Saon *et al.*, 2010].

Using the pronunciation dictionary, words are represented as sequences of phones which are modeled using 3-state left-to-right HMMs, without state skipping. All models have penta-phone cross-word acoustic context. The numbers of context-dependent states in the system is 4000, and a total of 200,000 Gaussian components. The acoustic models where initially ML trained and then discriminatively trained using the Boosted Maximum Mutual Information (BMMI) method.

11.4.2 Language Model

The system utilizes a 4-gram language model (of about 880 million n-grams) resulted from interpolating 20 language models trained with modified Kneser-Ney smoothing from the following resources: transcripts of the audio data, Arabic Gigaword corpus, and web transcripts for broadcast conversations collected by CMU/ISL (28M words from Al-Jazeera). The interpolation weights were optimized using GALE eval07 of BN and BC data which contains about 74K words. Note that the hypothesized dialectal data set described in Section 11.3.2 (DialectData) is not part of the data used for training these language models.

11.4.3 Decoding

The ASR decoding steps comprise the following: (1) The speech segments are clustered into speaker clusters; (2) These speech segments are decoded using a speaker independent system (3) Using the decoded output, speaker compensation transforms (VTLN, fMLLR and MLLR) are applied; and finally (4) the adapted segments are decoded again using the adapted models.

11.4.4 Pronunciation Dictionary

We saw in Chapter 3 that there are two main pronunciation modeling techniques for Arabic (unvowelized and vowelized). Recall that the unvowelized dictionary is created by running Buckwalter's morphology analyzer on every word in the vocabulary. Then, every undiacratized word is mapped to all the morphological analyses for which all the diacritics are removed (including the 3 MSA short vowels but not nunations). The vowelized dictionary is described in detail in Section 3.6.2.

We need to decide which dictionary to use for our ASR dialect experiments in this chapter. So, we train two identical ASR systems except for their pronunciation dictionaries, using the ASR settings described above, on BaseSet. The first system employes the unvowelized dictionary and the second utilizes the vowelized dictionary. The vocabulary size of both dictionaries is 795K words. Typically, a simple vowelized version of the dictionary performs better on MSA than a completely unvowelized version. To validate this, we test our two systems on GALE dev07, which consists of only MSA data. We find, in fact, that the WER using the unvowelized system is 12.4% but 11.4% using the vowelized system – the difference is statistically significant. However, testing these two systems on our dialect data (DialectTestSet), we find that the performance of the unvowelized system is significantly and consistently, across hypothesized dialects, better on this non-MSA data. The unvowelized system's WER is 46.8%, while the vowelized system achieves a WER of 48.2%. We believe that this is due to the fact that Buckwalter's morphology analyzer, which is used to obtain the short vowels, is designed to handle only MSA words, thus resulting in inaccurate pronunciations for dialectal data. From these results, we choose to use the unvowelized pronunciation dictionary for our experiments in this chapter.

11.5 Levantine-Specific Language Model

We test the hypothesis that optimizing the LM interpolation weights on hypothesized Levantine data will improve ASR on Levantine. Using the system design above, we train the system's acoustic models on BaseSet and compare its performance when employing the following language models:

1. Baseline LM-I: The LM described in Section 11.4.2 (without DialectData).

2. Baseline LM-II: This is a 4-gram language model results from interpolating: (1) The 20 LMs described in Section 11.4.2 and (2) a 4-gram LM trained on the hypothesized Levantine data set (LevTrain) (1.4 million n-grams). The interpolation weights are optimized utilizing GALE eval07 data set (same optimization set used for Baseline LM-I).

3. Levantine LM: Similar to the previous LM, but now we optimize the interpolation weights on our hypothesized Levantine dev set (LevDev). This is a Levantine-specific LM.

4. Levantine LM BC: This is a 4-gram language model results from interpolating: (1) The 20 LMs and (2) a 4-gram LM trained on LevBC (3.8 million n-grams). The interpolation weights are also optimized on LevDev. This is also Levantine-specific LM but with more noisy Levantine data (may contain MSA data).

Dialect	Baseline LM-I	Baseline LM-II	Levantine LM	Levantine LM BC
Levantine	**49.7**	**49.2**	**47.9**	**47.5**
Iraqi	49.5	50.1	48.8	49.0
Gulf	49.8	49.4	49.3	49.3
Egyptian	40.4	40.9	39.7	39.8

Table 11.3: Comparing the WER on each dialect using different LMs

We test these LMs on our hypothesized dialect test set (DialectTestSet) and report the WER in Table 11.5 for each of the four dialects. We observe that the Levantine-specific

LM slightly improves all dialects but as expected, significantly better (p-value < 0.001) with the most relative improvement on the hypothesized Levantine test set (LevTest). From Baseline LM-I, for Levantine, we obtain the greatest reduction in WER (2.2% absolute) when using the Levantine LM BC. Hence, adding the hypothesized Levantine BC data (although it is noisy) improves the WER by 0.4% over using only utterances tagged with non-MSA.

11.6 Levantine-Specific Acoustic Models

Dialect	Baseline AM	Levantine AM
Levantine	**56.1**	**54.3**
Iraqi	55.8	55.4
Gulf	56.3	55.1
Egyptian	45.8	44.6

Table 11.4: Comparing the WER on each dialect using Levantine vs. baseline Acoustic Models

In this section, we test the hypothesis that building Levantine-specific acoustic models (AM) may improve WER on Levantine input. For the Levantine system, we use all the 313 hours of speech from the hypothesized Levantine on BC (LevBC). The baseline acoustic models are trained on the 301 hours from BaseSet. Although both data sets have comparable number of hours, LevBC contains far fewer (almost half) speakers than BaseSet. This may bias the WER towards BaseSet. Since in this section we are only interested in the effect of clustering the acoustic data using dialect tags on the acoustic models, we use Baseline LM-I, described in previous section, for both systems.

Similar to the LM experiments, we compare the performance of these two systems on our hypothesized dialect test set (DialectTestSet). We report the WER for each hypothesized dialect in Table 11.6. We only report the ML-trained acoustic models in this section.[2]

[2] We accidentally deleted the BMMI-trained models and some important files needed to be able to retrain the models.

The system employing the Levantine-specific acoustic models performs significantly (p-value < 0.05) better (1.8% in absolute WER) than systems that utilize the baseline acoustic models on the Levantine test set (LevTest).

11.7 Levantine-Specific System

We have seen significant improvement in ASR when using dialect-specific language models and when using acoustic models separately. Now, we test the combination of these two types of dialect-specific models on (DialectTestSet). The baseline system's acoustic models are trained on BaseSet and its LM is Baseline LM-I. The dialect-specific system's acoustic models are trained on LevBC and its LM is Levantine LM BC. Both systems are discriminatively trained (using BMMI). Similar to previous experiments we compare the performance on our dialect test set (DialectTestSet). The WER for each dialect is shown in Table 11.7. We find that the Levantine-specific system achieves significant reduction in WER (of 4.6% absolute, 9.3% relative) over the baseline on the Levantine test set (LevTest). We also see a reduction in WER for all dialects, with the most improvement for Levantine, followed by Egyptian. The high improvement for Egyptian may suggest some similarities between the Egyptian and Levantine phonetic/allophonic structures. For example, both dialects typically replace the phoneme /q/ by a Glottal Stop. (See Chapter 2) Recall also that our acoustic data here (LevBC) is very noisy – it includes a great deal of MSA (only 27.3 out of the 313 hours were manually tagged as non-MSA). So, it could be that some of this MSA data is MSA-accented with Egyptian and other dialects.

Dialect	Baseline AM+LM	Levantine AM+LM
Levantine	**49.7**	**45.1**
Iraqi	49.5	47.6
Gulf	49.8	47.4
Egyptian	40.4	37.2

Table 11.5: Comparing the WER on each dialect using Levantine vs. baseline systems

11.8 Conclusions

We can conclude that targeting specific dialects and building specialized language models is better than simply treating the data as one cluster (i.e., leading to a single language model). We show that using our dialect ID system for clustering the transcripts and identifying the dialect prior to ASR improves WER. Similarly, training specialized acoustic models using speech data clustered using our dialect ID hypotheses improves results on the targeted dialect. When we employ both specialized types of models (LM+AM) for the ASR system, we obtain the best results (4.6% absolute; 9.3% relative) on the targeted dialect (Levantine in our case). These results also suggest that these two types of modeling are complementary to each other. This is not surprising since Arabic dialects differ phonetically and lexically. For future work, instead of building the Levantine acoustic models from scratch, we will experiment with MAP adapting already ML-trained acoustic models on MSA using our hypothesized Levantine acoustic data.

Chapter 12

Conclusions and Future Work

In this thesis, we have presented the results of a series of experiments in automatic dialect identification. We have compared previous to novel approaches, exploring a number of different features and modeling techniques for the task of dialect and accent recognition problem. We have experimented with lower and higher level information in the speech signal, including acoustic, phonetic, phonotactic, and prosodic information. We have developed a general approach for dialect and accent recognition that requires acoustic input only for training and testing. We have evaluated this approach on multiple dialects of different languages, and we have shown that, in most cases, we achieve state-of-the-art performance, even without combining different systems.

Most of the approaches we have employed in this work require a phone recognizer trained, ideally, on a variant of the language of the dialects to be identified. An essential component for building either a phone recognizer or a speech recognizer is a pronunciation dictionary. However, there has been no comprehensive study accurately modeling pronunciations in Arabic. In Chapter 3, we describe a method that addresses the difficulties inherent in modeling pronunciations for Modern Standard Arabic (MSA). In particular, we make use of a morphological analysis and disambiguation tool to predict most likely word diacritizations given the textual context of transcripts. We have developed a set of linguistically-motivated pronunciation rules and applied them on these diacritized words to automatically generate MSA pronunciation dictionaries. We have generated two different dictionaries: one is used during the training of the acoustic models and another is employed during the decoding phase in ASR. Using these dictionaries, we have improved the absolute phone error rate by 3.77%–7.29% over a state-of-the-art pronunciation model for Arabic. Also, we have obtained a 2.2% absolute significant reduction in Word Error Rate (WER) (11.5% relative) in a state-of-the-art Arabic ASR system.

We begin our work on dialect identification, we first analyzed the performance of well-known language recognition-based phonotactic approaches (PRLM and Parallel-PRLM) to distinguish four broad classes of Arabic dialects (Gulf, Iraqi, Levantine, and Egyptian) from each other and from MSA. Employing the Arabic phone recognizer described above, we have shown that these dialects significantly differ in terms of their phonotactic distributions. The Parallel-PRM approach can identify Arabic dialects with considerable accuracy, especially when employing a back-end classifier which combines the likelihoods from the n-gram models. Importantly, we have observed that Parallel-PRLM rarely confuses MSA with any other colloquial dialects, suggesting that MSA has its own distinct phonotactic constraints. Consistent with previous results in the language recognition literature, we have seen that Parallel-PRLM *significantly* outperforms the PRLM approach in almost all test-durations.

Next, we turned our attention to analyzing and modeling the prosodic structure of the four broad Arabic dialects. We have shown empirically that these dialects exhibit significant differences from one another in terms of the characteristics of their prosodic structure, including pitch range, register, and pitch dynamics, as well as differences in their rhythmic structure, speaking rate, and vowel durations. We have demonstrated that we can utilize these global prosodic features to automatically identify the dialect of a speaker with considerable accuracy. Modeling sequences of local prosodic features at the level of pseudo-syllables using HMMs significantly increases accuracy when combined with global prosodic features, resulting in an accuracy of 72.0% (chance is 25%). Such performance strongly suggests that prosody alone carries significant information for distinguishing these dialects. This finding is consistent with human perception studies of dialects of other languages, which have found that listeners can distinguish dialects using prosody. We have found that our prosodic modeling approach can also significantly improve a system that utilizes phonotactic features only, resulting in a classification accuracy of 86.3% on 2-minute-long utterances. While we have focused in this work on the prosody of Arabic dialects, our methodology should generalize to dialects of other languages. We have also observed that, the more difficult it is to distinguish a pair of dialects using the phonotactic approach, the more difficult it is to distinguish them using only prosodic features. It is possible that this correlation reflects an important underlying relationship between the prosodic and phonotactic structures of a dialect which may in fact constrain each other.

Based on the hypothesis that dialects differ in terms of their spectral distributions, we have also tested a standard acoustic-based approach, Gaussian Mixture Model – Universal Background Model (GMM-UBM), which has been widely employed by the speech and language recognition communities. We have found that this approach also performs well on our four Arabic dialects. We have improved this approach by applying a speaker adaptation technique to the feature space, i.e., feature space Maximum Likelihood Linear Regression (fMLLR). We compute the fMLLR transformation matrix using the first hypothesis of a context-dependent (CD) phone recognizer. We then use this matrix to transform the acoustic frames and then employ these transformed frames in the GMM-UBM approach. This feature transform significantly improves results — from an Equal Error Rate (EER) of 15.3% to one of 11.0%. We have also found that, consistent with the speaker recognition literature, Maximum-A-Posteriori (MAP) adapting only the Gaussian means for the UBM is sufficient also for dialect recognition.

None of the approaches described above has explicitly focused on subtle CD phonetic realization differences that may contribute to distinguishing the dialects of interest. To test the usefulness of such distinctions, we have introduced *Discriminative Phonotactics*, a novel approach to dialect recognition. This approach automatically identifies CD phonetic differences across dialects. In developing this approach, we represent each CD-phone with a supervector, a result of stacking the mean vectors of MAP-adapted GMMs of the corresponding CD acoustic model (HMM). We employ this novel representation to train SVM classifiers we use to *augment* the phonotactic sequences with dialect labels. We then train a second discriminative classifier to identify dialects. Thus, this Discriminative Phonotactics approach takes advantage of both phonotactic and acoustic-phonetic information.

Analyzing the performance of the Discriminative Phonotactics approach on detecting the four broad Arabic dialects, we have shown that it significantly outperforms PRLM

(EER of 17.3%) and GMM-UBM (EER of 15.3%) baselines as well as our own improved version GMM-UBM-fMLLR (EER of 11.0%). Discriminative Phonotactics achieves an EER of 6.0%, which represents an improvement of 5% in absolute EER and 45.5% in relative EER improvement over our best baseline (GMM-UBM-fMLLR). An important use of this framework is its ability to automatically extract linguistic knowledge — specifically, the phonetic cues that may distinguish one of our dialects from another. This system can be used to determine which phones in which contexts are realized differently across dialects.

Although Discriminative Phonotactics is quite effective in recognizing dialects, this approach has the disadvantage that it requires one to train a classifier for each CD-phone type for each pair of dialects. This can be quite expensive during training and recognition, and is potentially difficult to manage. To address this problem, we turned to kernel-based methods. Using the CD-phone supervector representation in Discriminative Phonotactics, we design a kernel function that cross compares CD-phone instances of the same type across pairs of utterances. Using this kernel function, we can train a *single* SVM classifier for each pair of dialects. This approach yields some (EER of 5.88%) but not significant improvement over the Discriminative Phonotactics approach. Employing CI-phones instead of CD-phones and using GMMs instead of HMMs, our kernel-based approach provides a significant reduction in EER (4.9%).

We observed that computing such a kernel function is quite time consuming, partly due to the cross comparison between our high dimensional supervectors. We resolved this problem by comparing supervectors of phone *types* as opposed to phone *instances* in our kernel function. To do that, we MAP-adapt each phone type's GMM-UBM using the acoustic frames of all the instances of this phone in the utterance. We thus obtain one supervector for each phone type. With this change, we obtain a substantial reduction in the time complexity of the algorithm (since results in a small constant number of comparisons), and even a slight reduction in EER (4.35%). Moreover, using an approximation of a KL-divergence (between phone-type GMMs) in our kernel function yields our best results — an EER of 3.96% on the four broad Arabic dialects. Finally, attempting to model features extracted from a larger context beyond CI-phones does not seem to yield significant gains.

We have also shown that, not only does the Kernel-GMM-Type-KL approach provide state-of-the-art results for four broad Arabic dialects, but it is also effective for distinguishing most pairs of Arabic sub-dialects, and it outperforms state-of-the-art approaches to recognizing American English vs. Indian English Accents. Moreover, we have seen that this approach outperforms the standard GMM-UBM approach for Southern vs. Non-Southern American English, a 53.5% relative improvement. This system also distinguishes a number of dialects from different American states plus Canada with considerable accuracy. It also achieves results comparable to a state-of-the-art Portuguese dialect recognition system, a system which, unlike ours, requires orthographic transcripts during training. These experiments lead us to conclude that our Kernel-GMM-Type-KL is general enough for languages other than Arabic. Finally, we have suggested a way to plot data-driven dialect maps using the pairwise EER between pairs of dialects.

We can conclude form all our experiments that phonetic features alone carry significant and nearly sufficient information to distinguish dialects. We have found that phonetic features modeled with our kernel-based methods substantially outperform all the approaches we tested in this thesis (previous and ours) that attempt to model phonotactics, prosodic and/or frame-based acoustic features (independent of phonetic constrains). Moreover, for our best performing approach (Kernel-GMM-Type-KL), we have seen that a speaker's utterance can be represented in a single vector which summarizes the general realization of the speaker's individual phones. It is important to note that, in our vector representation, the phone labels constrain which Gaussians can be affected by the MAP adaptation, i.e., the comparison in our kernel incorporates the linguistic constraints realized by the phone recognizer. This is in contrast to the previous GMM-supervector representation [Campbell *et al.*, 2006a] for which, in theory, any Gaussian in the GMM-UBM can be affected by any frame of any phone – ignoring the linguistic context of each frame. Empirically, we have found that our approach outperforms GMM-SVM for the six pairs of the broad Arabic dialects as well as for the American vs. Indian English task.

Finally, we have shown that, if we cluster the ASR acoustic training and testing data based on dialect labels using our dialect ID system and then train dialect-specific models, we improve ASR on Arabic. Specifically, we have found that optimizing the interpolation weights of a language model on a hypothesized Levantine development set (tagged by our dialect ID system) reduces the WER on a hypothesized Levantine test set. Similarly, training Levantine-specific acoustic models using speech data annotated as Levantine by our dialect ID system improves results on the Levantine test set as well. Combining both models to build a Levantine-specific ASR system, we obtain a significant reduction in WER (4.6% absolute; 9.3% relative) on Levantine. Employing hypothesized dialect IDs, we can conclude that using these dialect tags to target specific dialects and build specialized language and acoustic models is better than simply treating the data as one cluster, leading to a single language model and one set of acoustic models.

12.1 Summary of Contributions

This thesis represents the following contributions:

- Developing a new pronunciation modeling technique for MSA that significantly improves phone recognition as well as a state-of-the-art Arabic ASR system
- Providing thorough analyses of well-known methods (PRLM, Parallel-PRLM, GMM-UBM) used by the language and speaker recognition communities on Arabic dialects
- Improving the GMM-UBM approach using a speaker adaptation technique
- Proposing a new approach and features to model the prosodic structure of dialects and identifying prosodic structure differences across four broad Arabic dialects
- Developing two novel and general (i.e., language-independent) approaches for dialect and accent recognition that requires acoustic input only for training and testing
- Proposing a method to automatically identify phonetic knowledge (specifically, context-dependent phonetic cues) that contributes to our understanding of how dialects differ
- Being able to represent the summary of the phonetic content of any given utterance of any duration with a single vector of a fixed size (a vector of phone-type supervectors), a representation that we hypothesize can benefit multiple speech processing technologies (e.g., speaker verification)

- Achieving new state-of-the-art performance on most evaluated dialect and accent tasks
- Significantly improving ASR on Levantine using our dialect ID system

12.2 Further Work

We have seen that our best performing approach makes use of phone-type GMMs with a fixed number of Gaussians for all phones. In future work, we would like to experiment with data-driven methods to obtain the optimal number of Gaussians for each phone-type GMM. Also, we have observed that using GMMs for modeling phone-instances perform better than HMMs. We plan to test whether HMMs as opposed to GMMs perform better for modeling phone types. Note that HMMs have an advantage over GMMs since HMMs capture part of the phonetic temporal structure – HMMs would constrain even further the MAP adapted GMMs with sub-phonetic units (beginning, middle, and end of phones).

We have seen that our system outperforms Torres-Carrasquillo et al. [Torres-Carrasquillo *et al.*, 2008]'s system on Arabic and on American vs. Indian English. Their system employs a channel compensation component to remove language/dialect independent information and vocal-tract length normalization component to remove speaker-dependent information. Torres-Carrasquillo et al. showed that such components improved their results on both Arabic and English. In future work, we would like to test the impact of such techniques on our system's performance. Particularly, we are interested in experimenting with the SVM nuisance attribute projection (NAP) method [Campbell *et al.*, 2006b]. Although such a technique cannot be employed in a straightforward manner, due to the limited acoustic context of the phone-type supervector, we would like to experiment with compensating using the vector of all phone-type supervectors and then projecting this vector to a sub-space that removes channel effects, as in NAP.

The acoustic features we use for our non-Arabic experiments are RAST-PLP features with delta and delta-delta. We would like to experiment with discriminative features extracted from a large window of frames. In particular, we first will stack together N frames in one vector, and then derive an LDA matrix that projects this high dimension vector to lower dimensions where the classes for LDA are the dialects. This approach is widely used in ASR where phones are the classes. To our knowledge this method has not been employed for the language/dialect or speaker recognition.

We have mentioned above that using our phone-type supervector representation, an entire utterance can be represented as a single vector of phone-type supervectors. This representation can be viewed as the 'phonetic finger print' of the input speaker. We would like to test our system on the tasks of speaker identification and verification. In one of our preliminary experiments, we found that our kernel-based approach can identify the gender of the speaker with an accuracy of more than 98%. We would also like to augment this vector with prosodic features similar to those proposed in Chapter 6.

In ASR systems, the set of CD acoustic models is automatically generated by using phonetic decision trees which ask questions about the left and right contexts of each HMM state. Contexts with the smallest acoustic difference are clustered together. This procedure is necessary to maintain a balance between model complexity and the number of parameters to be robustly estimated from the training data. The questions that these trees ask are typically phonetic, such as: is the left=plosive and right=vowel?. Soltau et al. [2009] have found that extending the regular phonetic decision tree algorithm with dynamic questions about the "dialects" in the training and testing data improves the WER of the-state of-the-art-system in 0.6% (absolute). However, these dialectal questions were in fact simply annotations about the channel and program in the speech file, such as the Al-Jazeera Morning Show. The authors assumed that certain TV channels and channels are more likely to contain distinct dialects. In future work, we plan to use our system to automatically obtain dialectal annotations and then test this approach for improving the acoustic models.

Bibliography

[Abad *et al.*, 2009] A. Abad, I. Trancoso, N. Neto, and M. Ribeiro. Porting an European Portuguese Broadcast News Recognition System to Brazilian Portuguese. In *Interspeech*, Brighton, UK, 2009. 10.6.2

[Afify *et al.*, 2005] M. Afify, L. Nguyen, B. Xiang, S. Abdou, and J. Makhoul. Arabic broadcast news transcription using a one million word vocalized vocabulary. In *Interspeech*, page 16371640, 2005. 3.2

[Alorfi, 2008] F. S. Alorfi. PhD Dissertation: Automatic Identification Of Arabic Dialects Using Hidden Markov Models. In *University of Pittsburgh*, 2008. 4.2

[Ananthakrishnan *et al.*, 2005] S. Ananthakrishnan, S. Narayanan, and S. Bangalore. Automatic diacritization of arabic transcripts for asr. In *Proceedings of ICON*, Kanpur, India, 2005. 2.2.1

[Appen Pty Ltd, 2006a] Appen Pty Ltd. Gulf Arabic Conversational Telephone Speech – Linguistic Data Consortium, Philadelphia. Sydney, Australia, 2006. 4.3.1, 4.3.2

[Appen Pty Ltd, 2006b] Appen Pty Ltd. Iraqi Arabic Conversational Telephone Speech – Linguistic Data Consortium, Philadelphia. Sydney, Australia 2006. 4.3.1, 4.3.2

[Appen Pty Ltd, 2007] Appen Pty Ltd. Levantine Arabic Conversational Telephone Speech – Linguistic Data Consortium, Philadelphia. Sydney, Australia, Jan 2007. 4.3.2

[Barkat *et al.*, 1999] M. Barkat, J. Ohala, and F. Pellegrino. Prosody as a Distinctive Feature for the Discrimination of Arabic Dialects. In *Proceedings of Eurospeech'99*, 1999. 4.2

[Biadsy *et al.*, 2006] F. Biadsy, J. El-Sana, and N. Habash. Online Arabic handwriting recognition using Hidden Markov Models. In *IWFHR'10*, France, La Baule, 2006. 2.2

[Biadsy *et al.*, 2007] F. Biadsy, J. Hirschberg, A. Rosenberg, and W. Dakka. Comparing American and Palestinian Perceptions of Charisma Using Acoustic-Prosodic and Lexical Analysis. In *Interspeech*, 2007. 1

[Biadsy *et al.*, 2009] F. Biadsy, J. Hirschberg, and N. Habash. Spoken Arabic Dialect Identification Using Phonotactic Modeling. In *Proceedings of EACL 2009 Workshop on Computational Approaches to Semitic Languages*, Athens, Greece, 2009. 4

[Boersma and Weenink, 2001] P. Boersma and D. Weenink. Praat, a system for doing phonetics by computer. 2001. Software available at www.praat.org. 4.3.2

[Buckwalter, 2004] T. Buckwalter. Buckwalter Arabic Morphological Analyzer Version 2.0, 2004. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0. 1, 2.2.1

[Burget *et al.*, 2006] B. Burget, P. Matejka, and J. Cernock. Discriminative training techniques for acoustic language identification. In *Proceedings of ICASSP'06*, France, 2006. 4.2

[Campbell *et al.*, 2006a] W.M. Campbell, D.E. Sturim, and D.A. Reynolds. Support Vector Machines Using GMM Supervectors for Speaker Verification. *IEEE Signal Processing Letters*, 13(5):308–311, May 2006. 8.2, 8.2.1, 9.4.4.2, 12

[Campbell *et al.*, 2006b] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff. SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP variability compensation. In *Proceedings of ICASSP'06*, France, May 2006. 7.8, 9.4.4.2, 9.4.5, 10.3.2.1, 12.2

[Canavan and Zipperlen, 1996a] A. Canavan and G. Zipperlen. Callfriend american english-non-southern dialect. Linguistic Data Consortium, Philadelphia, 1996. 10.4.1

[Canavan and Zipperlen, 1996b] A. Canavan and G. Zipperlen. Callfriend american english-southern dialect. Linguistic Data Consortium, Philadelphia, 1996. 10.4.1

[Canavan and Zipperlen, 1996c] A. Canavan and G. Zipperlen. CALLFRIEND Egyptian Arabic Speech Linguistic Data Consortium, Philadelphia. 1996. 4.3, 4.3.1, 4.3.2

[Canavan and Zipperlen, 1996d] A. Canavan and G. Zipperlen. Callfriend hindi speech corpus. Linguistic Data Consortium, Philadelphia, 1996. 10.3.1

[Canavan and Zipperlen, 1996e] A. Canavan and G. Zipperlen. CALLHOME Spanish Speech – Linguistic Data Consortium, Philadelphia. 1996. 4.2

[Canavan *et al.*, 1997a] A. Canavan, D. Graff, and G. Zipperlen. Callhome american english speech. Linguistic Data Consortium, Philadelphia, 1997. 10.3.1

[Canavan *et al.*, 1997b] A. Canavan, G. Zipperlen, and D. Graff. CALLHOME Egyptian Arabic Speech Linguistic Data Consortium, Philadelphia. 1997. 4.3, 4.3.1, 4.3.2

[Chang and Lin, 2001] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. 2001. Software available at www.csie.ntu.edu.tw/ cjlin/libsvm. 3, 1

[Chen *et al.*, 2010] N.F. Chen, W. Shen, and J.P. Campbell. A linguistically-informative approach to dialect recognition using dialect-discriminating context dependent phonetic models. In *ICASP'10*, 2010. 4.2, 10.3.2.1, 10.3.2.2

[Cieri, 2005] C. Cieri. Fisher english training part 1 and 2, speech. Linguistic Data Consortium, Philadelphia, 2005. 10.3.1, 10.5.1

[Decker *et al.*, 2003] M.A. Decker, F. Antoine, P.B. Mareuil, I. Vasilescu, L. Lamel, J. Vaissiere, E. Geoffrois, and J.S. Lienard. Phonetic knowledge, phonotactics and perceptual validation for automatic language identification. In *ICPhS*, 2003. 3.1

[Do, 2003] M.N. Do. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Processing Letters*, 10(4):115–118, 2003. 9.4.4.2

[El-Imam, 2004] Y.A. El-Imam. Phonetization of arabic: rules and algorithms. In *Computer Speech and Language 18*, pages 339–373, 2004. 1, 2.2, 3.2

[Eskenazi, 1992] M. Eskenazi. Changing speech styles, speakers strategies in read speech and careful and casual spontaneous speech. In *Proceedings of the International Conference on Spoken Language Processing, Banff*, 1992. 1

[Frota and Vigrio, 2001] S. Frota and M. Vigrio. On the correlates of rhythmic distinctions: the European/Brazilian Portuguese case. *Probus*, 13, 2001. 10.6.2

[Gauvain and Lee, 1994] J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. In *IEEE Trans. Speech Audio Process*, volume 2, 1994. 1

[Habash and Rambow, 2005] N. Habash and O. Rambow. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. 2.2.1, 3.1, 3

[Habash and Rambow, 2007] N. Habash and O. Rambow. Arabic Diacritization through Full Morphological Tagging. In *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL07)*, 2007. 2.2.1, 3.1, 3

[Habash *et al.*, 2007] N. Habash, A. Soudi, and T. Buckwalter. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer, 2007. 2.2

[Habash, 2006] N. Habash. On Arabic and its Dialects. *Multilingual Magazine*, 17(81), 2006. 2.3.1

[Habash, 2010] N. Habash. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers, 2010. 2.1, 2.2

[Hamdi *et al.*, 2004] R. Hamdi, M. Barkat-Defradas, E. Ferragne, and F. Pellegrino. Speech Timing and Rhythmic Structure in Arabic Dialects: A Comparison of Two Approaches. In *Proceedings of Interspeech'04*, 2004. 4.2

[Hazen and Zue, 1993] T.J. Hazen and V.W. Zue. Automatic language identification using a segment-based approach. In *Eurospeech 93*, volume 2, 1993. 4.4, 4.4, 6.3, 8.1

[Hellmuth and El Zarka, 2007] S. Hellmuth and Dina El Zarka. Variation in phonetic realization or in phonological categories? Intonational pitch accents in Egyptian Colloquial Arabic and Egyptian Formal Arabic. In *Proceedings of 16th ICPhS*, 2007. 6.2.1

[Holes, 2004] C. Holes. *Modern Arabic: Structures, Functions, and Varieties.* Georgetown University Press, 2004. Revised Edition. 2.3.1, 5.5.3

[Kittler *et al.*, 1998] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On Combining Classifiers. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, 1998. 6.6

[Kneser and Ney, 1995] R. Kneser and H. Ney. Improved backing-off for n-gram language modeling. In *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1995. 3.6.1

[Koller *et al.*, 2010] O. Koller, A. Abad, and I. Trancoso. Exploiting variety-dependent Phones in Portuguese Variety Identification. In *Odyssey*, Brno, Czech Republic, 2010. 4.2, 10.6.1, 10.6.2

[Koller, 2010] O. Koller. Automatic Speech Recognition and Identification of African Portuguese. In *Diploma Thesis*, 2010. 10.6.1, 10.6.2

[Labov *et al.*, 1997] W. Labov, S. Ash, and C. Boberg. A national map of regional dialects of American English. In *Telsur Project, Linguistics Laboratory, University of Pennsylvania*, 1997. 10.5, 10.5.2

[Le, 2005] A. Le. 2005 nist language recognition evaluation. Linguistic Data Consortium, Philadelphia, 2005. 10.3.1

[Ma *et al.*, 2006] B. Ma, D. Zhu, and R. Tong. Chinese Dialect Identification Using Tone Features Based On Pitch Flux. In *Proceedings of ICASP'06*, 2006. 4.2

[Maamouri *et al.*, 2003] M. Maamouri, A. Bies, H. Jin, and T. Buckwalter. Arabic treebank: Part 1 v 2.0. Distributed by the Linguistic Data Consortium, 2003. LDC Catalog No.: LDC2003T06. 3.2

[Maamouri *et al.*, 2006] M. Maamouri, A. Bies, T. Buckwalter, M. Diab, N. Habash, O. Rambow, and D. Tabessi. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06*, Genoa, Italy, 2006. 4.3.1

[Mak and Rao, 2011] M.W. Mak and W. Rao. Utterance partitioning with acoustic vector resampling for GMMSVM speaker verification. *Speech Communication (2011)*, 53, 2011. 1

[Martin and Le, 2007] A. Martin and A. Le. 2007 nist language recognition evaluation test set. Linguistic Data Consortium, Philadelphia, 2007. 10.3.2

[Martin *et al.*, 1997] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *Proceedings of the European Conference on Speech Communication and Technology*, 1997. 4.5

[Matejka *et al.*, 2005] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil. Phonotactic Language Identification using High Quality Phoneme Recognition. In *Proceedings of Eurospeech'05*, 2005. 5.4, 10.3.2.2

[Matejka *et al.*, 2006] P. Matejka, L. Burget, P. Schwarz, and J. Cernocky. Brno university of technology system for nist 2005 language recognition evaluation. In *Proceedings of Odyssey*, 2006. 4.2, 7.5

[Messaoudi *et al.*, 2006] A. Messaoudi, J.L. Gauvain, and L. Lamel. Arabic broadcast news transcription using a one million word vocalized vocabulary. In *ICASP*, volume 1, page 10931096, 2006. 3.2

[Moreno *et al.*, 2004] P.J. Moreno, P.P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *Advances in Neural Information Processing Systems 16, MIT Press, Cambridge*, 2004. 9.4.4.2

[Murphy, 2004] K. Murphy. Hidden markov model (hmm) toolkit for matlab. In *www.cs.ubc.ca/ murphyk/software/hmm/hmm.htm*, 2004. 6.4

[Muthusamy *et al.*, 1992] Y. K. Muthusamy, R.A. Cole, and B.T. Oshika. The OGI Multi-Language Telephone Speech Corpus. In *Proceedings of ICSLP'92*, 1992. 5.4

[Ng and Jordan, 2002] A.Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, 2002. 8.4

[Ng, 2004] A.Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the 21 st International Conference on Machine Learning*, Banff, Canada, 2004. 8.4

[Nigam *et al.*, 1999] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden, 1999. 8.4

[Peters *et al.*, 2002] J. Peters, P. Gilles, P. Auer, and M. Selting. Identification of Regional Varieties by Intonational Cues. An Experimental Study on Hamburg and Berlin German. 45(2):115–139, 2002. 4.2

[Povey, 2004] D. Povey. Discriminative Training for Large Vocabulary Speech Recognition. In *PhD. Thesis, Cambridge University*, 2004. 4.2

[Ramus, 2002] F. Ramus. Acoustic Correlates of Linguistic Rhythm: Perspectives. In *Speech Prosody*, 2002. 4.2, 6.2.2

[Rennie and Dognin, 2008] S. Rennie and P.L. Dognin. Beyond Linear Transforms: Efficient Non-linear Dynamic Adaptation for Noise Robust Speech Recognition. In *ICASP'08*, 2008. 7.8

[Reynolds *et al.*, 2000] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19 – 41, 2000. 4.2, 7.4, 1, 7.6, 7.8

[Rosenberg and Hirschberg, 2006] A. Rosenberg and J. Hirschberg. On the correlation between energy and pitch accent in read english speech. In *Interspeech*, 2006. 6.3

[Rouas *et al.*, 2008] J.L. Rouas, I. Trancoso, M.C. Ribeiro, and M. Abreu. Language and variety verification on broadcast news for Portuguese. *Speech Communication (2008)*, 50, 2008. 10.6.2

[Rouas, 2007] J. Rouas. Automatic Prosodic Variations Modeling for Language and Dialect Discrimination. In *IEEE Transactions On Audio, Speech, and Language Processing*, volume 15, 2007. 6.2

[Saon *et al.*, 2010] G. Saon, H. Soltau, U. Chaudhari, S. Chu, B. Kingsbury, H.K. Kuo, L. Mangu, and D. Povey. The IBM 2008 GALE Arabic Speech Transcription System. In *Proceedings of ICASSP 2010*, Dallas, TX, 2010. 3.2, 3.6.2, 11.4, 11.4.1

[Shen *et al.*, 2008] W. Shen, N. Chen, and D. Reynolds. Dialect recognition using adapted phonetic models. In *Proceedings of INTERSPEECH*, Brisbane, Australia, 2008. 4.2

[Soltau *et al.*, 2007] H. Soltau, G. Saon, D. Povey, L. Mangu, B. Kingsbury, J. Kuo, M. Omar, and G. Zweig. The IBM 2006 GALE Arabic ASR System. In *ICASP*, 2007. 3.2, 3.6.2

[Soltau *et al.*, 2009] H. Soltau, G. Saon, B. Kingsbury, H.K.Kuo, L. Mangu, D. Povey, and A. Emami. Advances in arabic speech transcription at IBM under DARPA GALE program. *EEE Transactions on Audio, Speech and Language Processing*, 17(5):884–895, 2009. 3.1, 3.2, 2, 3.6, 3.6.2, 7.2, 7.7, 4, 12.2

[Stolcke, 2002] A. Stolcke. SRILM - an Extensible Language Modeling Toolkit. In *ICASP'02*, pages 901–904, 2002. 3.5.3, 5.5, 8.6

[Timoshenko and Hoge, 2007] E. Timoshenko and H. Hoge. Using Speech Rhythm for Acoustic Language Identification. In *Proceedings of Interspeech 2007*, 2007. 6.2, 6.3

[Torres-Carrasquillo *et al.*, 2004] P.A. Torres-Carrasquillo, T. P. Gleason, and D. A. Reynolds. Dialect identification using Gaussian Mixture Models. In *Proceedings of the Speaker and Language Recognition Workshop, Spain*, 2004. 4.2

[Torres-Carrasquillo *et al.*, 2008] P.A. Torres-Carrasquillo, D. Sturim, D. Reynolds, and A. McCree. Eigen-channel Compensation and Discriminatively Trained Gaussian Mixture Models for Dialect and Accent Recognition. In *INTERSPEECH*, Brisbane, Australia, 2008. 4.2, 7.6, 7.8, 9.6, 9.6, 10.3.1, 10.3.2.1, 10.3.2.2, 12.2

[Vapnik, 1999] V. Vapnik. The Nature of Statistical Learning Theory, 2nd edition. Springer Verlag. In *Telsur Project, Linguistics Laboratory, University of Pennsylvania*, 1999. 9.4.4.2, 10.4.2

[Vergyri and Kirchhoff, 2004] D. Vergyri and K. Kirchhoff. Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Ali Farghaly and Karine Megerdoomian, editors, *COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland, 2004. 2.2.1

[Vergyri *et al.*, 2008] D. Vergyri, A. Mandal, W. Wang, A. Stolcke, J. Zheng1, M. Graciarena, D. Rybach, C. Gollan 2, R. Schlter, K. Kirchhoff, A. Faria4, and N. Morgan. Development of the SRI/Nightingale Arabic ASR system. In *Interspeech*, 2008. 3.2, 2

[Wong and Sridharan, 2002] E. Wong and S. Sridharan. Methods to improve gaussian mixture model based language identification system. In *ICSLP*, 2002. 7.8

[Wong *et al.*, 2000] E. Wong, J. Pelecanos, S. Myers, and S. Sridharan. Language identification using efficient gaussian mixture model analysis. In *Australian International Conference on Speech Science and Technology*, 2000. 4.2, 7.6

[Wu *et al.*, 2004] T.F. Wu, C.J. Lin, and R.C. Weng. Probability estimates for multi-class classification by pairwise coupling. In *Journal of Machine Learning Research 5*, 2004. 9.2.3

[Young *et al.*, 2006] S. Young, G. Evermann, M. Gales, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. The htk book, version 3.4. 2006. 3.4, 3.5.1

[Zissman *et al.*, 1996] M.A. Zissman, T. Gleason, D. Rekart, and B. Losiewicz. Automatic Dialect Identification of Extemporaneous Conversational, Latin American Spanish Speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, 1996. 4.2

[Zissman, 1996] M.A. Zissman. Comparison of Four Approaches to Automatic Language Identification of Telephone Speech. *IEEE Transactions of Speech and Audio Processing*, 4(1), 1996. 4.2, 5.2, 5.3, 8.4

[Zitouni *et al.*, 2006] Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia, 2006. 2.2.1

Appendix A

Appendix

A.1 Pairwise American State Classification Results

State Pair	EER
Canada vs. NC	13.49
Canada vs. TN	15.43
Canada vs. IA	15.52
Canada vs. OK	16.30
Canada vs. LA	16.99
Canada vs. GA	17.04
AL vs. Canada	17.94
LA vs. WI	19.30
AL vs. IA	19.65
NC vs. WI	19.72
Canada vs. TX	19.81
Canada vs. OR	19.96
NJ vs. TN	20.01
NC vs. NJ	20.10
OK vs. WI	20.24
AL vs. CO	20.56
Canada vs. MD	20.60
AL vs. AZ	20.78
MA vs. NC	21.09
CO vs. LA	21.10
IA vs. NJ	21.11
TN vs. WI	21.55
GA vs. WI	21.98
AL vs. WI	22.17
CO vs. OK	22.24
LA vs. WA	22.65
LA vs. OR	22.69
NC vs. WA	23.13
AL vs. WA	23.19
IN vs. WI	23.51
Canada vs. OH	23.62
Canada vs. MO	23.65
NY vs. OK	23.92
NY vs. TN	23.97
LA vs. MN	23.99
FL vs. WI	24.06
MD vs. OR	24.34
NC vs. OR	24.43
CT vs. OK	24.55
NJ vs. OK	24.78
AL vs. MI	24.88
MD vs. NC	25.13
MN vs. NC	25.14
MN vs. TN	25.17
GA vs. NJ	25.32
MI vs. NC	25.45
IL vs. TN	25.65
OR vs. WI	25.75
Canada vs. VA	25.82
Canada vs. NJ	25.83
Canada vs. CT	25.85
CT vs. NC	25.99
CO vs. NC	26.05
MD vs. TN	26.15
MI vs. TN	26.19
TX vs. WI	26.23
GA vs. WA	26.29
Canada vs. MI	26.43
NJ vs. OR	26.45
Canada vs. IN	26.50
AL vs. OR	26.52
NC vs. NY	26.55
FL vs. IA	26.69
IA vs. VA	26.70
AL vs. MD	26.73
MA vs. TN	26.79
NJ vs. WI	26.79
MD vs. WI	26.86
IL vs. NC	26.94
IL vs. OK	27.09
AZ vs. Canada	27.15
Canada vs. SC	27.36
CA vs. NC	27.43
IA vs. TX	27.45

Table A.1: EER for Each Pair of American States

State Pair	EER
GA vs. MN	27.49
CO vs. TN	27.52
Canada vs. MA	27.93
AZ vs. WI	27.94
MN vs. TX	27.94
Canada vs. FL	27.94
MA vs. WA	27.94
AZ vs. NC	27.98
AL vs. MN	28.08
IA vs. NY	28.23
IA vs. WI	28.32
CO vs. IA	28.34
MN vs. OK	28.40
NC vs. PA	28.47
IN vs. MA	28.49
Canada vs. CO	28.54
Canada vs. IL	28.65
PA vs. WA	28.74
MN vs. NJ	28.80
IN vs. NJ	28.81
AZ vs. GA	28.83
MA vs. OH	28.86
IA vs. LA	28.89
CT vs. KS	28.95
IA vs. NC	28.98
TN vs. WA	29.10
GA vs. MA	29.12
MO vs. NJ	29.16
AL vs. PA	29.21
GA vs. OR	29.23
CA vs. CO	29.24
NY vs. OR	29.32
MA vs. OR	29.38
MI vs. TX	29.40
CT vs. OR	29.41
Canada vs. WI	29.41
NC vs. OH	29.44
CA vs. WI	29.46
LA vs. MI	29.48
LA vs. MD	29.50
PA vs. TN	29.63
MN vs. SC	29.64
AZ vs. TN	29.69
GA vs. MI	29.71
MD vs. MN	29.78
NJ vs. OH	29.82
MI vs. NJ	29.93
Canada vs. CA	30.03
Canada vs. PA	30.04
AL vs. MA	30.10
NJ vs. TX	30.17
IN vs. MN	30.25
IN vs. LA	30.34
SC vs. WI	30.57
AL vs. CA	30.62
OR vs. SC	30.68
AL vs. OH	30.76
NY vs. OH	30.80
CO vs. GA	30.93
LA vs. NJ	30.95
FL vs. MN	30.98
CO vs. FL	30.99
CA vs. GA	31.03
PA vs. WI	31.07
MA vs. TX	31.17
IN vs. NY	31.17
MI vs. NY	31.19
CT vs. LA	31.29
AZ vs. LA	31.29
AL vs. TX	31.35
GA vs. IA	31.36
IA vs. OK	31.38
NY vs. WA	31.41
CT vs. IA	31.41

Table A.2: EER for Each Pair of American States

State Pair	EER
IA vs. MD	31.44
MA vs. OK	31.44
MN vs. NY	31.47
Canada vs. WA	31.47
AZ vs. PA	31.49
MD vs. WA	31.50
AL vs. NJ	31.51
NY vs. WI	31.59
AL vs. NY	31.62
SC vs. WA	31.74
MN vs. PA	31.83
NJ vs. WA	31.83
CA vs. MI	31.88
MO vs. NC	31.88
MD vs. MI	31.92
GA vs. IL	32.03
FL vs. OR	32.16
OK vs. PA	32.17
Canada vs. NY	32.18
IN vs. OR	32.22
OH vs. WA	32.29
IN vs. NC	32.29
CA vs. NJ	32.30
AZ vs. MA	32.35
AZ vs. NJ	32.41
CT vs. TN	32.46
CT vs. MO	32.47
MO vs. WI	32.55
MA vs. SC	32.64
MI vs. OR	32.88
CA vs. IA	32.91
IA vs. MA	32.91
GA vs. NY	32.96
AL vs. OK	33.04
FL vs. WA	33.08
IL vs. NY	33.09
MD vs. MO	33.10
CO vs. WI	33.16
MA vs. WI	33.19
AZ vs. NY	33.21
PA vs. TX	33.21
IL vs. LA	33.26
LA vs. NC	33.34
KS vs. WI	33.36
CT vs. GA	33.37
FL vs. NJ	33.42
KS vs. LA	33.48
OK vs. WA	33.52
GA vs. OH	33.53
OR vs. TX	33.56
NJ vs. SC	33.56
TN vs. VA	33.60
OH vs. OR	33.61
Canada vs. MN	33.62
IA vs. SC	33.66
MA vs. MN	33.73
KS vs. NY	33.73
AZ vs. MN	33.87
CO vs. NJ	33.95
AL vs. IL	34.00
AZ vs. CO	34.01
CA vs. OH	34.04
CT vs. IN	34.06
MN vs. OH	34.06
CT vs. WA	34.07
IN vs. MD	34.09
TX vs. WA	34.10
CA vs. MN	34.11
CA vs. TN	34.12
CA vs. PA	34.13
MI vs. WI	34.16
OR vs. VA	34.17
CA vs. TX	34.18
MA vs. MI	34.37

Table A.3: EER for Each Pair of American States

State Pair	EER
CA vs. NY	34.39
IA vs. PA	34.42
GA vs. MD	34.49
IL vs. NJ	34.66
IL vs. TX	34.67
NC vs. TX	34.67
AL vs. MO	34.71
AZ vs. FL	34.76
MI vs. OH	34.82
MD vs. TX	34.82
CO vs. OR	34.84
AL vs. IN	34.86
GA vs. IN	34.88
AZ vs. IN	34.92
CA vs. MA	34.92
AZ vs. CT	34.94
LA vs. OH	34.95
FL vs. MI	35.09
IA vs. OR	35.09
IL vs. OR	35.10
NC vs. OK	35.13
GA vs. MO	35.16
GA vs. PA	35.17
IA vs. TN	35.22
CA vs. MD	35.25
IA vs. KS	35.30
CT vs. WI	35.34
CO vs. MA	35.37
MI vs. SC	35.51
MD vs. NJ	35.64
CA vs. OK	35.67
LA vs. PA	35.68
IN vs. WA	35.69
SC vs. TN	35.73
OH vs. WI	35.74
VA vs. WI	35.76
IA vs. MI	35.78
OH vs. TN	36.31
GA vs. TX	36.40
LA vs. NY	36.44
FL vs. MA	36.46
FL vs. NC	36.56
MI vs. MO	36.66
NY vs. PA	36.68
IA vs. OH	36.70
MO vs. NY	36.71
IL vs. PA	36.77
IN vs. MI	36.80
CO vs. MN	36.86
MD vs. OH	36.87
OR vs. PA	36.91
FL vs. NY	36.98
MA vs. VA	36.99
FL vs. GA	37.02
MO vs. OR	37.17
LA vs. MA	37.23
CA vs. FL	37.24
OH vs. OK	37.24
MA vs. PA	37.26
MD vs. OK	37.29
IL vs. MD	37.31
AL vs. FL	37.36
CT vs. OH	37.40
AZ vs. MD	37.40
OH vs. PA	37.42
IN vs. VA	37.48
AZ vs. WA	37.57
IL vs. WA	37.64
MD vs. SC	37.69
WA vs. WI	37.73
FL vs. PA	37.74
CO vs. NY	37.76
CO vs. MO	37.79
IL vs. VA	37.84

Table A.4: EER for Each Pair of American States

State Pair	EER
MN vs. MO	37.90
MN vs. VA	37.93
CT vs. MN	37.94
AL vs. VA	37.95
AL vs. CT	37.97
MI vs. OK	38.01
CA vs. WA	38.02
MI vs. WA	38.03
IA vs. WA	38.14
CA vs. IL	38.16
IN vs. OH	38.21
IN vs. PA	38.27
OH vs. VA	38.30
VA vs. WA	38.34
MA vs. MO	38.36
MD vs. NY	38.45
FL vs. TN	38.48
AZ vs. SC	38.50
CO vs. MD	38.54
IL vs. MA	38.68
MN vs. WA	38.82
AL vs. KS	38.95
IA vs. IN	39.00
IN vs. TX	39.01
CA vs. IN	39.09
MN vs. OR	39.21
AZ vs. MI	39.22
AZ vs. OH	39.23
MA vs. NJ	39.23
CO vs. MI	39.28
AZ vs. IL	39.28
IL vs. IN	39.29
GA vs. NC	39.32
FL vs. LA	39.42
CA vs. LA	39.46
NJ vs. VA	39.53
IL vs. OH	39.59
FL vs. IN	39.80
CT vs. TX	39.80
CT vs. MA	39.82
CA vs. VA	39.87
CO vs. WA	39.99
FL vs. OH	40.05
CO vs. IL	40.06
CO vs. TX	40.08
CO vs. PA	40.09
FL vs. OK	40.12
LA vs. TX	40.22
FL vs. IL	40.28
NC vs. VA	40.35
CT vs. MI	40.37
AL vs. NC	40.40
NY vs. VA	40.43
IL vs. SC	40.60
MA vs. NY	40.65
AZ vs. IA	40.69
MO vs. TN	40.83
IN vs. OK	40.85
IL vs. MO	40.93
MA vs. MD	40.95
CA vs. MO	40.96
GA vs. TN	41.16
MD vs. VA	41.27
NJ vs. PA	41.28
IA vs. MO	41.36
MN vs. WI	41.40
LA vs. MO	41.58
CO vs. SC	41.59
CO vs. CT	41.66
AZ vs. TX	41.74
FL vs. MD	41.75
CT vs. MD	41.75
KS vs. WA	41.76
KS vs. MN	41.78

Table A.5: EER for Each Pair of American States

State Pair	EER
MO vs. WA	42.08
LA vs. OK	42.35
LA vs. TN	42.37
MD vs. PA	42.56
IA vs. MN	42.71
PA vs. SC	42.78
MI vs. VA	42.82
AL vs. GA	42.87
GA vs. LA	43.03
IL vs. MN	43.11
AZ vs. OK	43.18
CO vs. IN	43.23
AZ vs. OR	43.29
AZ vs. VA	43.30
IL vs. WI	43.43
KS vs. SC	43.53
KS vs. NC	43.54
KS vs. VA	43.57
IL vs. MI	43.58
CT vs. SC	43.60
CO vs. KS	43.66
CT vs. NJ	43.76
CA vs. CT	43.76
CA vs. OR	43.88
CT vs. FL	44.10
CA vs. KS	44.23
AZ vs. KS	44.29
GA vs. OK	44.41
OK vs. TN	44.42
GA vs. VA	44.44
CT vs. PA	44.48
CT vs. VA	44.52
IN vs. MO	44.52
KS vs. NJ	44.60
AZ vs. CA	45.02
FL vs. MO	45.04
AL vs. TN	45.07
KS vs. MD	45.10
NJ vs. NY	45.13
OK vs. OR	45.16
KS vs. OK	45.18
KS vs. MA	45.19
FL vs. TX	45.24
GA vs. KS	45.29
LA vs. VA	45.30
NY vs. SC	45.59
MO vs. OH	45.61
AZ vs. MO	45.71
AL vs. LA	46.00
KS vs. TN	46.08
FL vs. SC	46.41
MO vs. SC	46.52
OH vs. SC	46.66
IN vs. KS	46.81
TN vs. TX	46.99
KS vs. PA	47.01
MO vs. OK	47.14
FL vs. KS	47.15
CA vs. SC	47.40
CO vs. VA	47.58
PA vs. VA	47.59
KS vs. MO	47.74
OK vs. VA	47.97
MO vs. VA	48.09
IN vs. SC	48.37
MO vs. TX	48.50
KS vs. OH	48.69
IL vs. KS	48.75
CT vs. IL	49.34
NC vs. TN	49.55
OR vs. WA	49.84
GA vs. SC	50.44
OK vs. TX	50.59
FL vs. VA	50.87

Table A.6: EER for Each Pair of American States

State Pair	EER
CT vs. NY	51.54
KS vs. MI	52.33
KS vs. TX	53.78
TX vs. VA	54.32
NC vs. SC	54.73
LA vs. SC	57.38
SC vs. VA	59.24
SC vs. TX	61.00
AL vs. SC	62.77

Table A.7: EER for Each Pair of American States