INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Improving Continuous Speech Recognition with Automatic Multiple Pronunciation Support

Charles Snow
School of Computer Science
McGill University

A thesis submitted
to the Faculty of Graduate Studies and Research
in partial fulfillment of the
requirements for
the degree of Doctor of Philosophy

December, 1997

Abstract

Conventional computer speech recognition systems use models of speech acoustics and the language of the recognition task in order to perform recognition. For all but trivial recognition tasks, sub-word units are modeled, typically phonemes. Recognizing words then requires a pronunciation dictionary (PD) to specify how each word is pronounced in terms of the units modeled. Even if the acoustic modeling component is perfect, the recognizer will still be prone to misrecognition, most often because the speaker can use a pronunciation other than that in the PD. This different pronunciation may be due to the speaker being a non-native speaker of the language being recognized, having 'mispronounced' the word, coarticulatory effects, recognizer errors in phoneme hypothesization, or any combination of these. One way to overcome these misrecognitions is to use a dynamic PD, able to acquire new pronunciations for words as they are encountered and misrecognized. The thesis examines the following questions: can automated methods be found that produce reliable alternate pronunciations? If so, does augmenting a PD (which originally contains only canonical pronunciations) with these alternate pronunciations lead to improved recognizer performance? It shows that using even simple methods, average reductions in word error rate of at least 45% are possible, even with speakers who are not native speakers of the recognition task language.

Resumé

Les systèmes de la reconnaisance de la parole traditionels utilisent des modèles acoustiques ainsi qu'un modèle pour la langue d'intêret lors de la reconnaissance. Ces systèmes peuvent modèliser différentes unités acoustiques : des mots entiers pour les tâches simples, ou des unités moins sophistiquées comme des phonèmes pour les tâches plus complexes. Ainsi, la reconnaissance des mots exige un dictionnaire des prononciations (DP) qui spécifie comment chaque mot est prononcé en utilisant les unités modèlisées. Même si la composante du système effectuant la modélisation acoustique est parfaite, le système demeure sensible aux erreurs de reconnaissance, surtout lorsque le locuteur utilise une prononciation autre que celle qui est décrite par le DP. Ceci peut arriver si le locuteur ne parle pas dans sa langue maternelle, si le mot a été malprononcé pour une raison quelconque, s'il y a eu des effets de coarticulation, si les phonèmes postulés sont inexacts, ou n'importe quel combinaison de ces causes. Une façon de surmonter ces erreurs de reconnaissance est d'employer un DP dynamique, qui est capable d'acquérir de nouvelles prononciations pour les mots mal reconnus. Cette thèse examine les questions suivantes: peut-on trouver des méthodes automatisées qui génèrent des prononciations alternatives fiables? Si oui, peut-on améliorer la performance d'un système de reconnaissance en ajoutant ces prononciations alternatives à un DP qui n'est basé que sur des prononciations canoniques? La thèse démontre qu'il est possible de reduire d'au moins 45% le taux d'erreur lors de la reconnaissance des mots par l'utilisation de méthodes simples et ce même si la langue maternelle du locuteur n'est pas celle d'intêret.

Contents

List of Figures iii
List of Tables v
List of Abbreviations vii
Acknowledgements ix
1. Problem and Proposed Solution 1
    1.1 Problem Overview 2
    1.2 Recognition Systems Past and Present 7
    1.3 Multiple Pronunciation as a Solution 10
    1.4 Thesis Organization 13
2. Speech Recognition 15
    2.1 Some Characteristics of Speech 15
        2.1.1 Acoustics of Speech 15
        2.1.2 Some Factors Affecting Speech 18
        2.1.3 Dialect and its Effect on Pronunciation 22
        2.1.4 Speaking Environment 23
    2.2 Acoustic Modeling 24
        2.2.1 Units to Model 24
        2.2.2 Features: Representing Acoustics 26
        2.2.3 HMM Based Acoustic Models 29
        2.2.4 Hidden Markov Models 33
        2.2.5 Evaluation 36
        2.2.6 Decoding 40
        2.2.7 Training 42
    2.3 Language Model 43
3. Multiple Pronunciations and Belief 51
    3.1 Rule Driven Approaches 52
    3.2 Non-Rule Driven Approaches 56
    3.3 Dialect and Accent Work 59
    3.4 Belief 60
4. Making and Assessing Variant Pronunciations 67
    4.1 Scoring 68
    4.2 An Obvious Suggestion 71
    4.3 Iterative Transformation 73
    4.4 The Phoneme Lattice 76
    4.5 A Substitution Rule Approach 77
    4.6 Rule Based Variant Generation 81
    4.7 Performance Models as Variant Generators 86
    4.8 Summary 90
5. Recognition Using Variant Pronunciations 93
    5.1 The Pronunciation Dictionary 93

5.2 Training 96
5.3 First Set 100
5.4 Second Set 101
5.5 Summary 106
6. Observations, Future Work and Conclusion 107
6.1 The Pronunciation Dictionary and Alternate Pronunciations 107
6.1.1 Adding to the PD 107
6.1.2 Generating Alternate Pronunciations 109
6.2 Pronunciation Likelihoods 112
6.3 Recognition 115
6.4 Preferred Pronunciations 116
6.5 Generalizing Alternate Pronunciations 116
6.6 Results 118
6.7 Future Work 120
6.8 Conclusion 122
Appendix 1: Speech Recognition Tasks and Corpora 125
A1.1 ATS2 Task and Corpus 125
A1.2 WSJ Task and Corpus 129
A1.3 TIMIT 130
Appendix 2: The Roger Speech Recognizer 133
A2.1 System Architecture 133
A2.2 System Components 133
A2.2.1 Recorder 133
A2.2.2 Feature Extractor 134
A2.2.3 Recognizer 135
References 137

List of Figures

Figure 1 Spectrogram of "Air Mexico" 6
Figure 2 Illustrative acoustic features of some phonemes 19
Figure 3 Illustration of an HMM 29
Figure 4 Isolated digit recognizer 37
Figure 5 Continuous digit recognizer 40
Figure 6 Segment of SFSA for airline callsigns 46
Figure 7 A causal network view of speech production 61
Figure 8 Selection of a pronunciation 62
Figure 9 Simple stochastic inference network 62
Figure 10 Simple HMM for deriving 65
Figure 11 Pronunciation Probability Across Dialectic Regions 70
Figure 12 Regional Variations in Substitution and Deletion of a Phoneme 71
Figure 13 Relaxation of misrecognized form toward canonical 73
Figure 14 Relaxation of misrecognized form toward canonical 75
Figure 15 Sample Phoneme Lattice showing activation of different phonemes 78
Figure 16 Relation of Number of Generated Variants to Length of Canonical Form 80
Figure 17 A phoneme performance model 87
Figure 18 Dialectic Variation in Candidate Probabilities 90
Figure 19 Output From a Training Run 99
Figure 20 Effect on Acoustic Score of a Coarticulation Pronunciation 113
Figure 21 Effect of Dialect Region on Pronunciation Likelihood 114
Figure 22 ATS2 Task Grammar 127

List of Tables

Table 1 Effect of introducing multiple pronunciations 14
Table 2 Phonemes used in North American English 17
Table 3 Optional rules whose probabilities are determined 54
Table 4 Initial Phoneme-by-Class assignments 79
Table 5 Variant Pronunciation Candidates Using Within-Class Substitution 80
Table 6 Lattice Filtering of Substitution-Rule Generated Variants 81
Table 7 Typical Protorules learned on wsj0 subset 85
Table 8 Variant Pronunciation Candidates Using Performance Model Generator 88
Table 9 Performance Model Substitution and Insertion Statistics 89
Table 10 Summary of Canonical Recognition Rates on ATS2 Task 96
Table 11 Successful Variant Pronunciations by Speaker 100
Table 12 Comparison of Rule-Based and Substitution Variant Generating Methods Success Rates 102
Table 13 Summary of Correction Rates on Training and Evaluation Sets 103
Table 14 Summary of Recognition Performance on Evaluation Set 104
Table 15 Summary of Recognition Performance on Test Set 105
Table 16 Effect of Mismatching Variants to Speakers 117
Table 17 Ubiquitousness of Variant Pronunciations 118
Table 18 Improved Test Set Results 120
Table 19 TIMIT phoneme units 132

List of Abbreviations

The following abbreviations appear in the thesis:

AM	Acoustic Model
ATIS	Air Travel Information System
ATS	ATS Aerospace Inc.
ATS2	ATS speech recognition task [task number 2]
CD	Context Dependent
CI	Context Independent
CDF	Cumulative Density Function
CPU	Central Processing Unit
CSR	Computer Speech Recognition
DARPA	Defense Advanced Research Projects Agency
DFT	Discrete Fourier Transform
dB	decibel
HMM	Hidden Markov Model
ICW	In Context Winner
kHz	kilohertz
LDC	Linguistic Data Consortium
LHS	left hand side [of a rule or equation]
LM	Language Model
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NRM	Naval Resource Management [task]
ms	millisecond
OAG	Official Airline Guide
PC	Personal Computer
PD	Pronunciation Dictionary
SER	String Error Rate
SFSA	Stochastic Finite State Automaton
SIN	Stochastic Inference Network
TIMIT	[task produced by] Texas Instruments and the Massachusetts Institute of Technology
VPC	Variant Pronunciation Candidate
VT	Vocal Tract
WER	Word Error Rate
WSJ	Wall Street Journal

Acknowledgements

I thank Providence that this thesis is now completed. So too, I am sure, do several others whom I also thank for their support and faith throughout the work. Foremost among these are my parents, who never wavered in their belief, though they came increasingly to be concerned if their longevity would extend to seeing the end of the thesis, and my wife, Leslie Daigle, who, more than anyone else, got to help me deal with the setbacks and successes along my way to thesis completion.

I am most deeply indebted to my thesis supervisor, Dr. Renato De Mori, for his help, guidance, support, suggestions, comments and generosity. Others whose help and suggestions were much appreciated (even if it may not have seemed so at the time) include Michael Galler, Philippe Boucher, Dr. Kostas Kontogiannis, Morris Bernstein, Jean Daigle, Ron Hall, and Sean Doyle.

I wish also to thank, in a particular way, for their roles in helping this thesis reach its conclusion, Prof. Chris Paige, who graciously helped with some of the editing, and Prof. Monty Newborn who, though peripherally involved, consistently provided concise and very meaningful observations.

Without the financial support I received, this work would simply have been impossible. Dr. De Mori was generous not only with his time and knowledge, but also with his financial support. Much of that support was provided by, in alphabetical order, ATS Aerospace Inc., the Institute for Robotics and Intelligent Systems (IRIS) and PRECARN Associates Inc., and the Natural Sciences and Engineering Research Council of Canada (NSERC). Other generous support was provided by companies for whom I worked on some small projects — they were most understanding of the problems pursuing doctoral research can introduce into work schedules. In particular, I wish to thank Caroline Singleton (then of Tordion Conseil) and Peter Deutsch of Bunyip Information Systems, Inc.

While they may appear last on my list, the systems staff of McGill's School of Computer Science, and the patient and long-suffering secretarial staff played heroic roles in keeping both the research and researcher up and running throughout the exercise.

To all of you, and others I may have neglected to mention specifically, I express my gratitude.

1. Problem and Proposed Solution

In spoken communication, a speaker communicates by forming a message and delivering it using an error-prone 'speech channel.' While the message may have been formulated with ideally pronounced words, the received spoken utterance may end up being less than ideal. A variety of factors, operating at different levels in the speech channel, may affect the message so that what is received is a distorted version of the original message. The research work described here is aimed at finding ways to improve the ability of computer speech recognition (CSR) systems to deal successfully with the distortions of ideal pronunciations which occur in real speech and are a source of difficulty, often leading to misrecognition.

Nearly all contemporary CSR systems are based on recognition of sub-word units, most commonly, of phonemes. Here, the recognizer must rely upon a pronunciation dictionary which, for each word known to the system, contains a pronunciation expressed in the hypothesized sub-word units. It is here that a weakness is introduced into the recognizer, i.e., the dependence upon a pronunciation dictionary with, almost ubiquitously, a single pronunciation for each word.

The weakness is that to the extent that a given speaker pronounces words in the same way the pronunciation dictionary prescribes, the recognizer may be expected to work well. But where the speaker uses a different pronunciation, either erroneously, or as the result of the speaker's dialect, or because the language being used is not the speaker's native language, misrecognition increases with the disparity between what the speaker spoke and what the pronunciation dictionary led the recognizer to expect. This becomes a serious problem hampering the widespread deployment of speaker independent recognition systems, particularly in cases where speakers are not expected to have any particular training beforehand, or be required to honour specific syntax constraints.

An obvious suggestion is to use multiple† pronunciations (canonical plus variants) for words in the pronunciation dictionary. What is not obvious is what specific variants to use. One reason for this is that variants introduced to accommodate a particular speaker or group of speakers (e.g., with Austrian-accented English) may not be suited to accommodate a different speaker or group of speakers (e.g., with Pakistani-accented English). A set of such variants may, in fact, induce errors beyond what the canonical pronunciations alone would have provided in terms of recognition accuracy. Given, though, that the use of multiple pronunciations has been found useful (see section 1.3), is there a procedure for identifying which multiple pronunciations it is helpful to add? Can such a procedure, if it is found, be automated?

The research work described here examines these questions, concluding that it is possible to automate – if not fully, then nearly so – such a procedure and, as a consequence, improve recognition accuracy for small vocabulary tasks. The methods proposed can achieve average reductions in both word and string error rates of more than 45% on an air traffic control task.

1.1 Problem Overview

Let *WS* be a word string to be recognized (e.g., transcribed into text). *WS* is formulated by a speaker, then uttered using the speaker's vocal tract and articulators to produce sounds communicating *WS*. The stream of acoustic events comprising the utterance travels through air (at least) to a listener who interprets them as, ultimately, words of an utterance. The process of reconstructing *WS* from this stream of acoustic events is recognition, in the minimal sense (i.e., no interpretation or consequent action).

† Also called 'alternate' pronunciations in the literature. The terms will be used interchangeably in this thesis.

One obvious way for a computer to recognize speech would be to store all possible word strings and then identify which one corresponded to a particular utterance. This is infeasible as natural spoken languages can generate a countably infinite number of such strings.

A second suggestion would be to store all words W and by concatenating sequentially recognized individual words conveyed by the acoustic events, recognize WS. This approach has been exploited with some success, but is ultimately limited due to the problem of variability in the acoustic events representing the individual words and their juxtaposition.

Indeed, this problem of variability is the essence of the speech recognition problem. The acoustic events representing a single word, spoken in isolation of any other word, will never be identical across multiple utterances, even for one speaker. This problem becomes monumentally worse when one strives to perform recognition of speech which is:

(1) *continuous*: words spoken without unnatural pauses between them, as is the case in isolated word speech,

(2) *speaker independent*: irrespective of the age, gender, or dialect of the speaker, recognition must be correctly performed,

(3) *large vocabulary*: for most real-world problems, a recognizer must work correctly with a vocabulary of several (≥ 10) thousand words.

Recognizer performance is subject to further requirements, such as:

(4) *accuracy*: human users have exceedingly high expectations for the accuracy of a system,

(5) *speed*: correct recognition must be rendered in real time, i.e., at the speed the speaker speaks.

Successful recognition systems are used today which do not handle all of the above effectively. For many specialized applications, acceptable performance may be achieved with recognizers that handle only one or two of the above points. But in the case of more general purpose applications (and particularly where users are not specially qualified), it is difficult to envision successful deployment of speech recognition technologies which do not deal with the above problems in a realistic way.

An inspired — and now the dominant — approach to the recognition problem [4] uses probabilities as follows: let the stream of acoustic events in an utterance be represented as the *observation* $\mathbf{Y} = y_1, y_2, \cdots, y_T$ for an utterance of duration T. The speech recognition problem can then be simply formulated as finding the word string $\hat{WS}$ having the highest probability given the observation $\mathbf{Y}$:

$$P(\hat{WS}|\mathbf{Y}) = \max_{WS} P(WS|\mathbf{Y})$$

$$= \max_{WS} \frac{P(WS)\,P(\mathbf{Y}|WS)}{P(\mathbf{Y})}. \qquad (1)$$

For a given observation $\mathbf{Y}$, the optimal $\hat{WS}$ can be found without knowing $P(\mathbf{Y})$. The two remaining probabilities, $P(WS)$ and $P(\mathbf{Y}|WS)$ are estimated from a *language* model (LM) and an *acoustic* model (AM), respectively.

The details of how these two models are constructed and used may be found elsewhere (see Chapter 2). For the present it suffices to say that most recognizers actually recognize sub-word units, most often *phonemes*. In order to produce meaningful recognition this string of phonemes must then be rendered as the string of words *WS*. To do this, one must have some dictionary-style mapping between each word and the phoneme string which corresponds to its pronunciation(s).

One might assume that if the string of phonemes hypothesized by the above recognizer (i.e., the string having the highest global probability amongst all possible strings suggested by $\mathbf{Y}$) is perfect — corresponds exactly and unfailingly to what the speaker really spoke — that recognition would similarly be perfect. This is not the case.

Firstly, a speaker may have used a valid pronunciation which does not occur in the recognizer's pronunciation dictionary (PD). For example, the word "via" may be pronounced† /v iy ae/ or /v ay ae/. The word "zero" may be one of /z ih r ow/, /z iy r ow/, or /z ih r ix/. Recognition systems typically only support one pronunciation per word.

Secondly, a speaker may have mispronounced a word. The resulting stream of uttered phonemes is faithfully recognized as the (incorrect) phoneme string for a word. This incorrect string will either not appear at all in the PD, or, worse, appear as the correct pronunciation for some other word. An example is a native English speaker's pronunciation of the airline name "Air Mexico". The spectrogram of Figure 1 makes it clear that the speaker pronounced the name as:

/eh r m ih kcl k s kcl k ow/,

where the expected /ih/ in the canonical

/m eh kcl k s *ih* kcl k ow/

is totally absent.

Thirdly, another form of mispronunciation unique to continuous speech recognizers arises from between-word *coarticulation*, where the last phoneme of word i and the first phoneme of word $i+1$ interfere with each other. For example, in the string "6 17", the word final /s/ of '6' and the word initial /s/ of '17' may be merged by the speaker – only a single /s/ is spoken, resulting in /s ih kcl k *s* eh v eh n tcl t iy

† A table describing the notation for phonemes used throughout this thesis is found in Table 2.

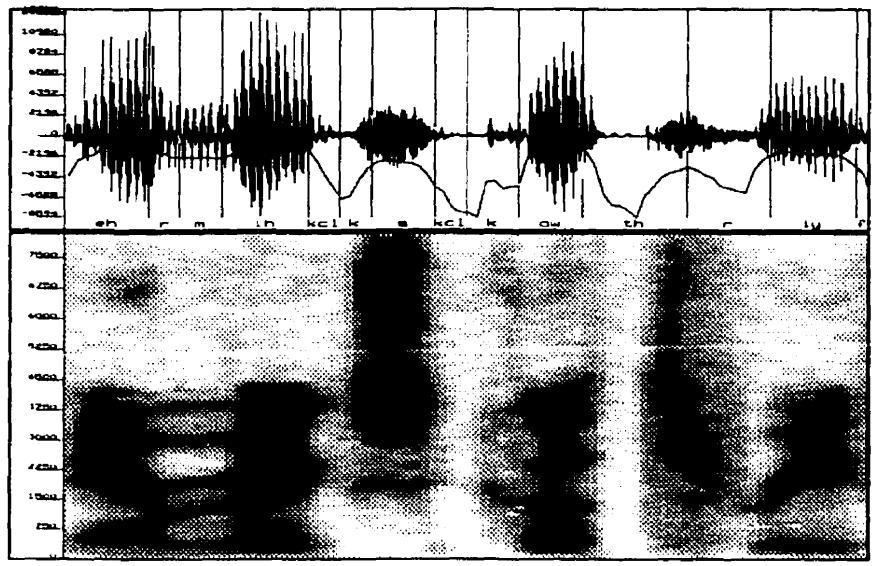

Figure 1 Spectrogram of "Air Mexico" The airline name is pronounced by an American native-English speaker and is very clearly pronounced 'correctly' to a human listener. The spectrogram shows, with equal clarity, that the phoneme ih, expected in m eh kcl k s *ih* kcl k ow is missing entirely.

n/. In other cases the effect is more dramatic, e.g., in "did you", the word final /d/ of 'did' and word initial /y/ of 'you' often merge into the single affricate /jh/ resulting in /dcl d ih jh uw/.

A fourth problem is that natural speech contains non-speech events which human listeners ignore, but which are awkward for recognition systems to cope with. Such events include throat clearing, coughs, pauses, "um" noises, corrections and restarts.

Finally, the phoneme string hypothesized by the recognizer will, in reality, be imperfect, thereby adding recognition errors to the above problems. Recognition errors may be due to any or all of the following: weaknesses in the recognizer itself, factors in the acoustic environment (e.g., background noise, especially speech from other speakers), flaws in the transduction of the acoustics to the electrical signal, conversion to digital form, signal analysis, etc.

Extensive on-going work in the speech recognition research community aims to improve robustness, accuracy, power and flexibility of recognition systems. While improvements in acoustic modeling, where much of this effort is directed, can be expected to yield improvements in recognizer performance, even perfect acoustic modeling cannot assure perfect recognition. As described in § 2.2, establishing the parameters of these models is time consuming and costly. There is somewhat of a "diminishing marginal return" quality to continued work on these models, as increasing amounts of person and computer time eke out small gains in recognizer performance. Before describing a different area in the recognition process where even modest effort may improve performance (§ 1.3), let us take a brief look at what is currently available in the way of CSR systems.

1.2 Recognition Systems Past and Present

The ultimate goal of speech recognition technology is to produce a system with which spoken interaction is as effortless as it would be with another human. That is, it would need to provide 'human quality' performance in the five categories listed in the previous section, as well as satisfying:

(6) *system training*: users cannot be required to 'train' ('enroll') the system to recognize their speech before using it for recognition,

(7) *user training*: users need not be instructed in how to interact with the system (e.g., requiring a user to pause between each word),

(8) *natural speech*: users do not have to adopt a particular syntax (beyond that of the language being used) in order to interact successfully.

There is also the pragmatic consideration that the recognition system must have an implementation that both renders it readily affordable and permits it to fit well with existing, familiar suites of applications.

Speech recognition, as a technology, can be split into two main categories: (i) systems where an exact transcription of the spoken utterance is required, and, (ii) systems where the primary concern is acting upon the meaning of a spoken message. Note that in this latter case, exact recognition is not necessary (and may be undesirable) so long as the content of the utterance is correctly interpreted. A variant of the first category is word- or topic-spotting systems, which 'listen' to a stream of speech, but recognize only particular words identified as 'of interest' either themselves or in that they relate to a topic. The first category is normally split on vocabulary size, with large vocabulary systems referred to as *dictation* systems.

A common application of contemporary small vocabulary recognition systems is control and command of some system, where spoken input supplements traditional input modalities. This may include simple one word commands, or digits, or both. Current accuracy of single digit recognition (11 word vocabulary, with perplexity† 11) where the length of digit string is known, is reported to be 0.3% word error rate [66].

The research community has established standardized recognition tasks that allow direct and meaningful comparisons to be made of different recognition systems. The ATIS (Air Travel Information System) task is designed to assess performance on "...spontaneous goal-directed..." [23] natural language spoken queries related to air travel. It originally featured a vocabulary of under 1,500 words. From the spoken utterance is derived a query which is processed on a standard subset of the OAG (Official Airline Guide) database. Assessment of the recognizer is based on the correctness of the result of the query. Recent word error rates reported on this task are between 2 and 3% [14].

† *Perplexity* is a measure of the average number of word choices possible from a given word, and is dependent on the task grammar. For digit recognition, any digit can follow any other, hence the perplexity is equal to the vocabulary size. For a more detailed description of perplexity see section 2.3.

For assessment of dictation systems, a popular task is the Wall Street Journal (WSJ) task [44,48]. This task, consisting of many components (see Appendix 1), provides utterances that are read text from the Wall Street Journal newspaper. These texts are divided into those using a 5,000 word vocabulary, and those using one of 20,000 words. Word error rates reported for the 5,000 word task are typically in the 9 to 10 % range [19,38] though some report higher rates (16%) [36].

Another task popular in the late 1980's and early 90's was the Naval Resource Management (NRM) Task created by the U. S. Defense Advanced Research Projects Agency (DARPA). This task, with a vocabulary size of 967 words, accepted natural language queries concerning naval vessels and resources. A typical sentence is "Is Rathburne located in Wellington or Aberdeen?" One of the notable recognition systems to emerge from that period was Kai Fu Lee's SPHINX system [32], developed at Carnegie Mellon University, which provided a best word error rate of 6.3%. SPHINX is noteworthy also in that it replaced in its lexicon the canonical form of each word with the most frequently observed form. While the gain in accuracy was reportedly a 27.4% improvement in WER, SPHINX still provided only one pronunciation per word.

A growing number of CSR systems are now commercially available. Currently, several vendors have isolated-word systems, and a growing number are now releasing continuous speech recognizers. Whereas CSR systems used to be expensive, stand-alone devices — general purpose computers of the day lacked the power to perform reasonable recognition — contemporary systems are almost universally software packages that work with the standard CPU and memory of a host computer (e.g., a PC, Macintosh, or workstation) and use the computer's standard audio hardware to deal with audio input.

Small vocabulary, template-based systems, suitable for command and control functions, but not for general dictation, and intended for speaker-dependent use, are available for under $100.

Dictation systems with vocabularies of over 10,000 words, isolated-word or continuous, are available for under $1,000 and provide rather good performance. Word accuracy rates (percent words correct) quoted by vendors, and by independent reviewers appearing in promotional material, are invariably in the high 90s. It is not uncommon, however, to see such systems perform best on native American-English speakers, and less well as a speaker's manner of speaking (i.e., dialect or accent) diverges from this 'standard.'

1.3 Multiple Pronunciation as a Solution

Of the many areas within CSR where research activity is on-going, one area that, until very recently, appears to have been largely overlooked is that of the pronunciation dictionary. The research work described here targets this area specifically by examining several methods for generating and managing multiple pronunciations for each word. The objective is to improve a PD dynamically, making use of real data from the recognizer as it operates. The benefits of multiple pronunciations include:

(1) accommodating multiple valid pronunciations for words, eliminating the first of the abovementioned (§ 1.1) four problems.

(2) accommodating multiple 'invalid' pronunciations of a word, largely mitigating the second of the abovementioned four problems. Primarily this point addresses cases where a large number of users pronounce a word differently than the PD as a result of dialect, e.g., the Austrian city name 'Klagenfurt' will be pronounced by a native English speaker from North America differently than by a native Austrian who is a non-native English speaker. This can also be used to correct some major coarticulation errors.

(3) known, consistent errors in the recognizer could be compensated for at the level of the PD, and generally for lower overall cost than repairs elsewhere in

the recognizer (e.g., retraining of acoustic models; see § 2.2.7).

The case for the beneficial effects of multiple pronunciations on recognition rates is now fairly well established [28,30,53,60,65] – and not just for English [25,57], [32] notwithstanding. In each of these cases, the multiple pronunciations were either determined by application of rules suggested by phoneticians, or trained on a corpus so that the recognizer has multiple pronunciations of 'all' words. While it may be concluded that support for multiple pronunciations is desirable, this support is not, by itself, adequate to provide overall improved recognition. This is true for several reasons. Firstly, a system which allows multiple pronunciations but cannot provide the actual variants to be added in an informed way is of little value. Secondly, a system which allows blind introduction of variants for any case of misrecognition quickly becomes unusable because the PD becomes bloated with variants; incorrect recognition of acoustically confusable words eventually increases as more and more variants are introduced [10].

The work described here not only examines support for multiple pronunciations, but also the managing of these pronunciations. Suppose that the uttered string $WS = W_1 W_2 \cdots W_i \cdots W_n$ is incorrectly recognized as $WS = W_1 W_2 \cdots W_j \cdots W_n$, where W_j is incorrectly recognized for W_i. Upon being notified of this error, the system should:

(1) generate a small, plausible set of variant pronunciations for W_i and test these so as to obtain a measure of each variant's suitability; generating variant pronunciations is done using knowledge about English pronunciation, either provided *a priori* or observed from on-going recognition and/or some corpus of utterances, as well as using 'knowledge' obtained from previously introduced variant pronunciations,

(2) for those variants which correct the misrecognition, determine which one or ones will be introduced into the PD,

To do this, the system begins with a dictionary of ideal (canonical) pronunciations of words and expects users' pronunciations of words will be *distortions* of those in this dictionary. The origins of these distortions can be modeled using a probabilistic mechanism (belief networks [46]) which integrates nicely into the overall stochastic recognition approach.

Belief networks provide a means of modeling the way in which one thing influences another (e.g., the way a speaker's choice of phonemes influences the pronunciation of the word) and the way in which observed evidence can affect the belief in something, measured as a probability (e.g., the way acoustic evidence can affect the belief a particular word distortion was spoken).

In brief, our original approach may be summarized as follows. In recognizing an utterance, suppose word W_i has been misrecognized as W_j. Let w_i represent the canonical pronunciation of word W_i, and w_r the phoneme string hypothesized for W_i[†]. With w_i we can generate, using some mechanism, a set of distortions w_{i_k} among which is w_r, or, a group of one or more pronunciations that are 'close' enough to w_r that use of any one of them leads to correct recognition of W_i.

To correct the misrecognition, we wish to add to the set of canonical pronunciations in the PD. In order to know which of the pronunciation candidates w_{i_k} to add, we select the one having the maximum $Bel(w_{i_k})$, the belief that w_{i_k} is a distortion of W_i. We show (in Chapter 3) that this belief can be determined as:

$$Bel(w_{i_k}) = \alpha P(\mathbf{Y}|w_{i_k})\, P(w_{i_k}|W_i). \qquad (2)$$

The acoustic modeling component of the recognizer provides $P(\mathbf{Y}|w_{i_k})$, the probability that uttering w_{i_k} 'explains' the observed acoustic description $\mathbf{Y}$. Ways of estimating $P(w_{i_k}|W_i)$ are developed.

† w_r may, in fact be w_j, in which case the misrecognition will be difficult to correct. It is much more likely that w_r is a phoneme string intermediate between w_i and w_j but 'close' to w_j, hence the (mis)recognition of W_j for W_i.

By using this approach, we are also able to ensure that (1) the system can infer generalizable qualities from corrections it has applied, so that it does not accumulate a large number of pronunciations for words, and, eventually, (2) a correction observed to be successful in a particular context could be more broadly applied (automatically) to similar pronunciation contexts, hopefully avoiding future misrecognitions of other words.

Clearly, not all such generalizable qualities will prove to be correct, so a means for retracting an application must be provided as well.

Pragmatically, the 'generalizable qualities' are represented as rules. Positive justification for a rule occurs whenever a variant pronunciation suggested by application of the rule results in correct recognition. Similarly, negative justification occurs when misrecognition is due to the use of the rule-derived variant pronunciation.

The use of multiple pronunciations was expected to be particularly valuable in situations where non-native speakers of English have to use an English language recognizer. Preliminary work [40] bore out this expectation: see Table 1. These results, from use of manually introduced multiple pronunciations on a small number of sentences, were intended to serve as a demonstration of the utility of multiple pronunciations, and inspired the more formal investigation described here.

1.4 Thesis Organization

The remainder of this thesis provides more details on the mechanics of both acoustic and language models, and describes each of the following problems encountered when using multiple pronunciations: how to generate them, how to assess them, and how to utilize them effectively. Chapter 2 provides background information on

Table 1 Effect of introducing multiple pronunciations: The table shows the effect upon word error rates in an air traffic control task of introducing multiple pronunciations over the use of purely canonical pronunciations. Two of the speakers were native English speakers (1 and 3); two non-native English speakers had very pronounced accents (Greek (2) and Italian (5)). The other speaker (4) is a native French speaker. Note not only the reduction in overall error rate, but also the compression of the range of word error from 60.7 to only 12.3 (80% reduction) [40] .

	Error Rate		%
Speaker	Canonical	Multiple	Improvement
1	19.9	1.3	93.3
2	70.0	11.9	83.0
3	26.9	9.4	64.9
4	39.9	10.0	74.7
5	80.6	13.6	83.1

speech and speech recognition, describing the acoustic modeling methods currently in widespread use, as well as the language modeling component of a recognition system. Chapter 3 looks more particularly at the role of multiple pronunciations in a recognizer, at how other work in the area uses them, and describes a metric that can be used to assess the quality of particular variant pronunciations. Chapter 4 discusses several methods that can be used to generate multiple pronunciations, and Chapter 5 presents results of using these methods. The thesis concludes with Chapter 6, which discusses the effectiveness of the use of the different methods of generating multiple pronunciations, the strengths and weaknesses discovered, and suggests future work which can be based on the work presented here. Two appendices follow: the first describes the recognition tasks discussed in the thesis, the second presents the recognition system used for all experimental work.

Results are shown for an air-traffic control task developed for a local company.† It has a vocabulary of about 100 words with average perplexity of 3.3. The nature of the task, recognizing air traffic control sentences, allows the use of a fairly rigid syntax.

† ATS Aerospace Inc., St. Bruno, Québec

2. Speech Recognition

The speech recognition problem may be formulated as follows. For an observation string $\mathbf{Y} = y_1, y_2, \cdots, y_T$ representing the stream of acoustic events for an utterance of duration T, the selection of the sequence of word hypotheses $\hat{WS}$ is chosen, using Bayes' formula, as:

$$P(\hat{WS}|\mathbf{Y}) = \max_{WS} \frac{P(WS)\, P(\mathbf{Y}|WS)}{P(\mathbf{Y})}.$$

$P(\mathbf{Y}|WS)$ is estimated using an acoustic model to provide the posterior probability of the observation given the word string. $P(WS)$, estimated by a language model, is an independent measure of the probability of WS assuming some language constraint. While the latter is not strictly necessary to a recognizer, e.g., one can set all words to being equiprobable, the improvement in recognition accuracy brought about by the constraining power of a LM is dramatic. This chapter provides a brief introduction to the general theory underlying acoustic and language modeling.

2.1 Some Characteristics of Speech

Before looking at how the acoustics of speech are modeled, let us briefly review the acoustics of speech, and phenomena which can lead to misrecognition. This review is not meant to be comprehensive, but focuses on background material relevant to the thesis work.

2.1.1 Acoustics of Speech

These acoustics are generated by air flow through the (1) vocal tract (VT: throat, nasal cavity, mouth), (2) articulators (lips, teeth, tongue, vellum), and, (3) vocal folds. The vocal folds, located in the larynx, can impart an oscillatory excitation (voicing)

to the system, by opening and closing in the outgoing air flow. Thus the system may be thought of as a set of connected tubes or cavities of variable size and shape (vocal tract), in which are located movable objects (articulators) which can direct air flow and, in so doing, introduce turbulence. The air flow through these cavities, around and over the articulators, may be accompanied by an oscillatory excitation or not.

The motion of the articulators is continuous, involving the coordinated activity of over 100 muscles, hence there is the potential for a very large range of different sounds to be generated. In examining sounds related to speech it is common to group them in terms of the articulatory features from which they arise. In so doing, approximately 60 distinct groupings of sounds, called *phonemes*, are identified, distinguished by three articulatory differences: place of articulation, manner of articulation, and the presence or absence of voicing. Each spoken language uses some subset of these phonemes, generating words by concatenating the appropriate phonemes; North American English uses approximately 40 of them (see Tables 2, 19). Based on their acoustic characteristics, phonemes may be grouped into the classes:

1. vowels and diphthongs,
2. fricatives and affricates,
3. nasals,
4. glides and liquids,
5. stops

Each class has fairly distinctive qualities, e.g., vowels being characterized by strong voicing as opposed to stops which have a typical silence followed by a burst of energy. The voicing observed in vowels is the contribution made by the excitation of the vocal folds oscillating in the outward air flow, at a frequency F0, with resonances imparted by the vocal tract. These resonances are numbered beginning with the resonance having the lowest frequency, i.e., F1, F2, F3 and occasionally F4, and are

Table 2 Phonemes used in North American English: The table shows the place and manner of articulation for phonemes used in North American English. The place of articulation is the point at which the vocal tract is most constricted. For vowels, the nature of the constriction is also relevant, and so the tongue position and rounding of the lips is used in describing place of articulation. The tongue body may create the greatest constriction at Front, Middle, or Back of the mouth cavity; further, it may be High (close to roof of mouth cavity) or Low. The lips may be Rounded in forming the vowel, or not. Vowels requiring the tongue to move far from a central, neutral position are designated Tense; those having closer places and/or shorter durations, are called laX. In some vowels, the place of articulation moves during pronunciation of the vowel; these appear as starting → ending places. Notes: (1) retroflex (the tongue tip curls upward and backward), (2) schwa (a weak mid vowel). – appears where no letter label applies. Table adapted from [39].

Phoneme	Example Word	Articulation Place	Manner	Voiced
iy	sEAt	H F T	V	+
ih	sIt	H F X	V	+
ey	sAte	M F T	V	+
eh	sEt	M F X	V	+
ae	sAt	L F T	V	+
aa	sOt	L B T	V	+
uw	sOOn	H B T R	V	+
uh	sOOt	H B X R	V	+
ow	sOAk	M B T R	V	+
ao	sOUght	M B X R	V	+
ah	sUb	M B X	V	+
er	sERve	M – T (1)	V	+
ax	sofA	M – X (2)	V	+
ix	sunkEn	H B X	V	+
ay	sIte	L B → H F	D	+
oy	sOY	M B → H F	D	+
aw	sOW	L B → H B	D	+
y	You	front unrounded	G	+
w	Woo	back rounded	G	+
l	Loo	alveolar	L	+
r	Rue	retroflex	L	+
m	Moo	labial	N	+
n	New	alveolar	N	+
ng	siNG	velar	N	+
f	Fox	labiodental	F	–
v	Vox	labiodental	F	+
th	THief	dental	F	–
dh	THey	dental	F	+
s	Sue	alveolar strident	F	–
z	Zoo	alveolar strident	F	+
sh	SHock	palatal strident	F	–
zh	meaSure	palatal strident	F	+
hh	Hay	glottal	F	–
p	Pox	labial	S	–
b	Box	labial	S	+
t	Talks	alveolar	S	–
d	Docks	alveolar	S	+
dx	siTTer	alveolar	S	–
k	Cox	velar	S	–
g	Gawks	velar	S	+
ch	CHocks	alveopalatal	A	–
jh	Jocks	alveopalatal	A	+

referred to as *formants*. Formants depend on the shape of the vocal tract during the pronunciation of the given phoneme, and are quite distinctive from one vowel to the next; see Figure 2.

A distinction is made between the phoneme, as an atomic speech unit different from other such units, and the *phone* which is simply a speech sound. One may think of the phoneme as an idealized form of what a particular speech sound should be, whereas the phone is the spoken, acoustic realization of a phoneme. Since the VT and articulators represent a continuous system, many different phones realizing a phoneme are possible.

2.1.2 Some Factors Affecting Speech

Speech may be thought of as a stream of phonemes generated by a speaker. Since articulatory movement is continuous, this stream is not a 'clean' sequence of clearly distinct phones. Rather, the articulators move smoothly from one phone configuration, 'through' the next (i.e., briefly assuming the configuration for that phone) and thence to (through) the following phone, and so on. Consequently, the i^{th} phone's configuration can affect that of at least its immediate predecessor (and successor). In nasals, for example, the vellum tends to lower before the constriction in the oral cavity is complete, and be raised after the oral re-opening, leading to nasalization of the phones on either side of the nasal [42]. Another example is the sequence "did you" in which the /y/ of 'you' and the second /d/ in 'did' tend to merge (palatization of /d/), resulting in /d ih jh uw/ rather than the anticipated /d ih d y uw/ [11].

The rate at which the speaker speaks can influence the character of the speech. As the rate increases, the time a phone remains in a steady state decreases. The articulators tend not so much to move faster as move less (i.e., over shorter distances) [20]. Increases in rate have little effect on F0 and formants [62], but can affect intelligibility by reducing durational contrasts that help cueing of phonemes

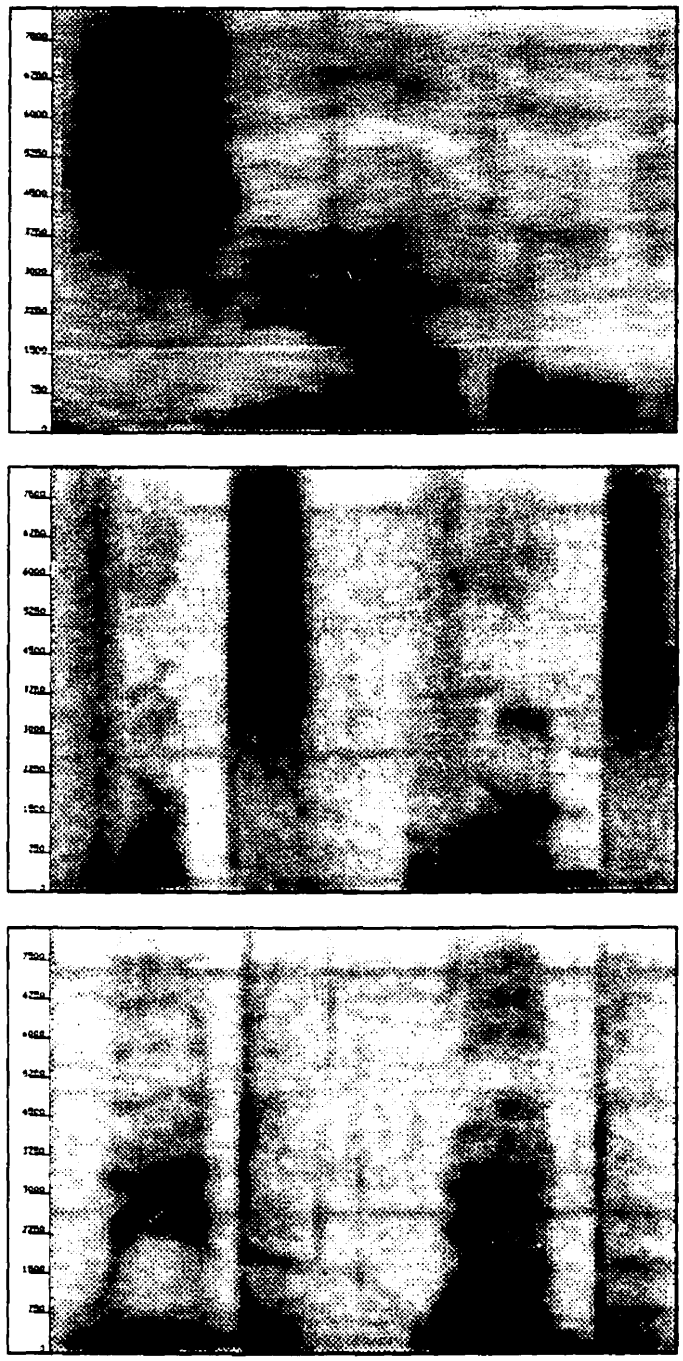

Figure 2 Illustrative acoustic features of some phonemes The first spectrogram shows the word "Seattle" being spoken in isolation; total duration shown is 507 ms. Intense high frequency frication marks the word initial /s/. The following /iy/ is marked by four high energy bands that are parallel throughout (at approximately 125 (F0), 312 (F1), 2125 (F2) and 2812 Hz (F3)); the lowest two are merged into a single dark band in this spectrogram, separating only in the /ae/ following the /iy/, with F1 rising to approximately 562 Hz while F0 remains relatively fixed at 125 Hz. The formant that in /iy/ appeared at 2125 Hz drops through the /ae/ to about

1400 Hz. The stop for the /t/ is very clear, and the follow on /ah l/ shows the F1 and F2 converging to approximately 600 Hz. F3 stops at the /t/ closure and is not present thereafter. Note that the trajectories of these two formants towards their final positions began before the stop and continues after it.

The second spectrogram shows the nonsense phrase "forts vords" spoken as an isolated pair of words; duration is approximately 1270 ms. Note the frication with which these words begin is different from that observed in "Seattle", and is different in each word (in part since the /v/ is voiced whereas /f/ is not). The central portion in each, though identical, appears different in both duration and formant behaviour. Note, too, the difference in the stops of the /t/ and the /d/.

The third spectrogram shows the words "weed wed" spoken as an isolated pair; duration is approximately 1150 ms. This pair of words serves primarily to showcase the difference between a 'long' and 'short' vowel sound: /iy/ in 'weed' as opposed to /eh/ in 'wed'. As in the first case above, note that trajectories of formants in 'weed' begin a drop toward a word final position before the stop of /d/ and continue afterward. The analogous trajectory in 'wed' is a fixed intensity, held level throughout the remainder of the word.

[50]. This translates into increased recognition errors — two to four times the word error rate of average speakers [43] due to increases in deletions and substitutions, when the phoneme rate is as little as one standard deviation above the mean rate [56]. Slower than normal speech consists mostly of increased pauses [16] and so differs little from normal rate speech.

Another factor which affects speech is whether the speaker is conversing naturally (spontaneous speech) or is reading a text aloud (read speech). While there may not be appreciable effects on the spectral qualities of individual phonemes, there are qualitative differences in the overall speech, including:

- rate is more consistent in read speech; spontaneous speech is characterized by frequent between-word pauses, often punctuated by sounds like "um", "ah",
- volume is more consistent in read speech,
- annunciation is typically clearer for read speech.

All of the above descriptions assume a speaker speaking at a 'normal' volume. Very low volume speech (whispering) is different in that the voicing excitation

provided by the vocal folds is replaced with frication at the glottis (the same effect which produces /h/). Phonemes which are normally voiced are now much weaker in amplitude, often sounding less loud than fricatives; phonemes which are normally unvoiced are unaffected.

Shouted speech is loud primarily because the speaker is forcing more air through the VT, which is more open than in normal speech, raising F1 (by 43 to 113 Hz, and F2 as well in female 'shouters' [27]), increasing amplitude, and affecting the spectral properties of vowels, glides, liquids and nasals. F0 is also raised and vowels are lengthened in shouted speech. The changes in relative intensities of formants make vowel sounds more intense but less distinctive. The distortion of formant frequencies can have a significant effect on recognition performance; one study reported a 34% drop in recognition accuracy with respect to unshouted [34].

A third variant of ordinary speech is singing. Singing differs from ordinary speech in that duration of phonemes (usually vowels rather than consonants) are modified to suit musical rhythm requirements, and F0, rather than varying continuously, is held fixed for periods of time corresponding to the musical notes being sung. Singers face the same problem as 'shouters' when they need to sing loudly, namely, increased air flow through the glottis tends to raise F0, which the singers are trying to control more rigidly. In overcoming this problem, singers lower the larynx in vowels [7], resulting in the introduction of a 'singer's formant' in the range of 2300 Hz to 3200 Hz [54] boosting by roughly 20 dB energy in the F3/F4 range. The presence of this formant appears to be key to a singer being heard over an accompanying orchestra since there is otherwise little significant difference in the maximum intensity of voice between trained and untrained voices.

2.1.3 Dialect and its Effect on Pronunciation

Speakers of a language pronounce a word by pronouncing a concatenation of phonemes corresponding to the word. A number of factors, described above, can influence the sound of the phonemes (generation of different phones) and resulting word. Another factor which influences the sound of words is *dialect.*

A dialect is a distinctive form of a language which, while intelligible as the particular base language, differs in any or all of pronunciation, vocabulary and grammar. Dialects can usually be attributed to a particular ethnic or social group, or to a particular region.

Of prime relevance to the work described here are differences due to pronunciation. Such differences result from the use of a 'distinctive' phoneme string in pronouncing a word either through use of one (or more) different phones and/or the insertion or deletion of phones. For example, the words 'marry merry Mary' tend all to be pronounced as /m eh r iy/ throughout most of the U. S. , particularly so on the west coast, except in the east (especially the northeast) where they are pronounced distinctly as /m ae r iy/, /m eh r iy/, /m ah ry y/ [1].

Another regional example is a dialect found in the New York City area characterized by aberrant use of /r/. For some words, the /r/ is dropped completely, e.g., 'car', 'four.' Other words have inserted /r/ sounds at the end, e.g., 'idea' becomes /ay d iy r/, 'saw' becomes /s aa r/. The /r/ appearing above represents the effect of merging /ah/ with an /r/ which immediately follows it, producing a new sound (an "r-colored vowel" [1]).

When a speaker speaks a language other than its native tongue, dialect-like effects (in pronunciation) are imparted to the non-native tongue. One of these effects occurs when phonemes used in the non-native tongue do not exist in the speaker's native tongue. This becomes a problem since after a certain age the

speaker can no longer easily learn how to generate new phonemes, and may be unable to perceive a phoneme as being distinct from ones in the native tongue, e.g., Japanese has no /l/ and native Japanese speakers usually report perceiving such a sound as /r/.

2.1.4 Speaking Environment

A speaker never speaks in absolute isolation from influences of its surroundings. The speaker's environment can have a dramatic effect on the recognition accuracy of speech.

The first contribution made by the surrounding environment is echo and reverberation. Assume a speaker in an ordinary room speaking to either another person or into a recognition system, i.e., there is a particular location in the room at which the speech is being observed. The acoustic signal reaching that observation point is a combination of what emanated from the speaker's VT, as well as acoustical energy reflected from floor, ceiling, and walls of the room. There can be quite complex interactions of constructive/destructive interference between the 'direct' speech with the multiple reflections. This situation can be made much worse by allowing the speaker's position with respect to the observation point to vary, since there will cease to be a fixed pattern of distortion introduced by the constructive and destructive interference of reflections. Also, different types and sizes of room will have different effects: a carpeted room will tend to deaden high frequency reflections compared to a room with a hardwood floor.

Independent of room 'colouration' of the speech signal, there may occur *noise*. Noise can be steady (e.g., ventilation fan) or bursty (e.g., 'click' of a door closing). Most office environments feature steady white noise (energy evenly distributed across entire spectrum) arising from building and equipment ventilation, some hum (60 Hz [in North America] and multiples thereof) from electrical and fluorescent

lighting, as well as burst noise such as telephones ringing, doors opening and closing. Further, there is typically speech from other speakers in the vicinity of the observation point, and this mixture of multiple-speaker speech is highly variable.

For analysis and discussion purposes, the amount of noise is quantitatively reported either as an intensity level, e.g., 90 dB, or as the signal-to-noise ratio:

$$SNR = 20 \log_{10}\left(\frac{\text{average energy in noise-free signal}}{\text{average energy in noise corrupting the signal}}\right) dB$$

Room colouration and the cacophony of noise which may accompany it can have serious degradatory effects on recognition systems. Low frequency components in the noise can be mistaken for voicing, thereby 'converting' an unvoiced phoneme into a voiced one, e.g., confusing a /b/ for a /p/.

Speech recognition systems may find their way into less hospitable environments than offices. In moving automobiles, noise levels are typically much higher, and the SNR can be < 5 dB [47]. Speech carried over telephone systems is band-limited to a pass band of roughly 4 kHz (thus particularly affecting the intelligability of fricatives), and may be subject to amplitude compression and other spectral distorting effects.

On factory shop floors noise levels may be high, with burst noise figuring prominently. Jet fighter cockpits can reach noise levels exceeding 90 dB. A speaker will be forced to shout in either of these noisy settings, adding distortion due to the Lombard effect to the growing difficulty of recognizing the speech.

2.2 Acoustic Modeling

2.2.1 Units to Model

Because phonemes are relatively few in number, and since any word in a language can be generated as a concatenation of them, phonemes are an obvious choice as the

acoustic units to model. In modeling phonemes one may choose to model context dependent (CD) or context independent (CI) phonemes. In the case of the latter, there is one acoustic model for each distinct phoneme, and training such models is straightforward: one instance, in any context, of a phoneme is an instance of training data for the model. But, as context exerts a great influence on the articulation of a phoneme, these CI models remain sensitive to context. A large amount of training data, presenting the phoneme in many different contexts, is hence necessary.

Since the CI model needs to capture so much contextual variability, and this variability influences the beginning and ending but not much the middle of the phoneme, models with richer modeling power for begin and end sections are sometimes used.

Context dependent models aim to provide much more accurate modeling of acoustics by providing a model for each context in which a modeled phoneme may appear. In principle this would require n^3 models, n being the number of phonemes to be modeled as:

<left phoneme context> target_phoneme <right phoneme context>

i.e., a *triphone*. Even though the majority of these triphones never occur, e.g., many combinations are not physiologically possible, while others never occur in the language whose phonemes are being modeled, there remains a very large number of triphones to train (typically several hundred or a few thousand [32]). Very large training corpora may still only provide an insignificant number of occurrences of a triphone, hence a clustering technique may be used to provide a 'generalized triphone'.

An intermediate strategy between triphones and context-independent units that still captures some context effects is the use of a *diphone* model.

One may also choose to use 'word dependent phones', where a particular phoneme model is trained in the context of a particular word. While not suitable for large vocabulary tasks, for small vocabularies or small subsets of words which may be difficult to recognize, e.g., function words (like 'a', 'the', etc.) these models may be

of benefit [32]. Various combinations of these phoneme models may be used.

One need not, of course, choose the phoneme as the acoustic unit to model. Acoustic modeling of whole words is rarely done since there may be tens of thousands to be modeled for a task. Cases where whole word acoustic modeling is successfully used are typically very small vocabulary tasks, e.g., spoken digit recognition [51], and simple command function tasks with a small set of imperatives, e.g., "yes", "no", (e.g., as in Nortel's Automated Alternate Billing Service), or "stop", "go", "left", "right", "up", "down", etc.

Other possibilities include linguistically inspired units like syllable, demi-syllables or pseudo-syllabic segments, as well as acoustically motivated units like fenones [9].

In this thesis, unless mentioned otherwise, it may be assumed that any mention of units being acoustically modeled refers to context independent† phonemes.

2.2.2 Features: Representing Acoustics

This section does not aim to discuss the relative merits of various sets of features which can be derived from acoustics for the purposes of recognition, but rather to describe why features are computed, and what features are to be used for experimental work during the research.

The stream of acoustic energy carrying speech must, to become accessible to software manipulation, be converted first to an analog electrical signal, and then to digital form. Any reasonable quality microphone has a response curve which assures that the range of frequencies where speech energy occurs is transduced with

† While greater recognition accuracy would be expected by modeling context *depdendent* units, such models were unavailable with the recognition system used in experimental work reported on in this thesis.

adequate linearity. This signal is sampled, typically at 16,000 samples/second; the sampled version thus correctly represents components of the speech signal up to 8 kHz in frequency. Sample values are represented linearly in 16 bit signed integers.

The feature extraction step, performed next, is intended to reduce the amount of data by selectively representing characteristics of the signal relevant to speech. Samples are grouped into frames, typically of 20 ms. duration and a set of *acoustic features* is computed for each frame. The features are selected so as to provide information useful in identifying what phonemes may be present in the part of the signal corresponding to that frame. Commonly used features include the overall energy in the frame, E, and its first time derivative, ΔE: these are particularly useful in identifying stop consonants.

The spectrum of the signal — the distribution of energy with respect to time and frequency — along with its first and second time derivatives, conveys information useful in identifying phonemes, e.g., the time course of formants through a vowel, or the presence of high frequency frication noise in a fricative.

In computing spectral information from the speech signal, one first determines the frequency domain representation of the signal, computed using a discrete Fourier transform, DFT, applied frame by frame. With this representation, one usually attempts to emulate the behaviour of the ear's basilar membrane so as to extract spectral information that would be perceptually relevant for the recognition of speech. Perceptual tests on humans demonstrate that the individual component frequencies of a complex sound cannot be distinguished if all components lie within a particular bandwidth. To be distinguishable, a component frequency must lie outside of this *critical bandwidth* [37]. Critical bandwidths are reported to be 10% to 20% of the sound's center frequency [49]. This feature suggests that one can mimic the basilar membrane's characteristics using a series of triangular filters with center frequencies spaced one critical bandwidth apart.

A common choice for the spacing of these filters follows the *mel* scale, which aims to transform the signal's frequency scale into a perceptually meaningful and linear scale. This is most often done by spacing the filters linearly at frequencies below 1 kHz, and logarithmically at higher frequencies.

Now one computes the energy in each filter. One way to do this would be to provide the inverse DFT of the value in each filter. Another involves viewing the overall speech signal as being the result of a convolution of an excitatory source (periodic pulses or noise) and a linear, time-varying system (vocal tract). Were it possible to de-convolve these two signals, one would have useful information about how the speech was generated, in particular, whether the speech is voiced or not, and if so, a good basis for estimating the fundamental period of the voicing. This can be done (for speech) from the signal spectrum by taking its logarithm. The result of using such a homomorphism is that the originally convolved signals are now additive, hence the effects of each can be linearly separated following an inverse transform. In practice, following the log computation, the real *cepstrum* [49] is computed by a discrete transform, e.g.,

$$c_d(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log|X(k)|e^{\frac{j2\pi kn}{N}}, n = 0, 1, \cdots, N-1 .$$

In the case where mel-scaled filters have been used, each cepstral coefficient is computed using the energy value in each filter. Thus, the spectrum of the speech signal is reported as a set of n mel-scaled cepstral coefficients.

The total number of features computed per frame is typically anywhere from 25 to 50. Thus, a frame of speech 20 ms in duration (16 kHz sampling × 16 bits/sample = 640 bytes) is now represented by a feature vector of length 100 to 200 bytes. In addition to data reduction, the features capture the essence of the signal based upon physiological and acoustical analysis. It is this set of features, chosen to represent the speech acoustics, that the acoustic models will model.

2.2.3 HMM Based Acoustic Models

The Hidden Markov Model† (HMM) has been found to be highly successful for modeling the acoustics of phonemes; see Figure 3.

Briefly, the model consists of states and transitions. The model generates an output value, observable from 'outside,' whenever a state change (to the same or some other state) occurs. If the output values are chosen from the domain of observations of the object being modeled, then the HMM may be viewed as generating a sequence of observations.

In addition to the above, the HMM has two sets of probabilities. One set is associated with the transitions: each transition from a state has a probability of being the transition chosen. The other set is associated with each of the possible values generated as output when taking a transition. The effect is that a generated sequence of observations has a certain probability associated with it resulting from the combined products of transition probability and output value probability for that transition for all transitions taken.

The models are called "hidden" since there may be many possible state sequences which can result in a particular output sequence, but observing the model

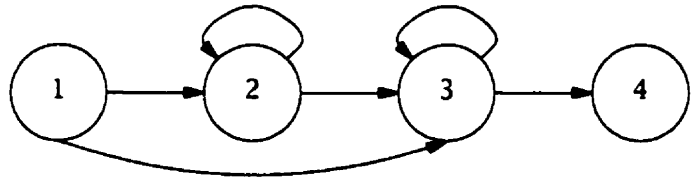

Figure 3 Illustration of an HMM having 4 states, 6 transitions (all left → right)

† This section provides an elementary overview of how a Hidden Markov Model is used to model speech. Readers familiar with HMMs may wish to proceed directly to the following section.

from the outside, it is not possible to know which state sequence was, in fact, used.

For example, in the HMM in Figure 3, let us suppose that the event being modeled is coin tossing, i.e., the values output by the model are either 'H' or 'T' and that there are the following transition and symbol probabilities:

Transition			Symbol Probability	
From	To	Probability	H	T
1	2	0.6	0.5	0.5
1	3	0.4	0.9	0.1
2	2	0.3	0.2	0.8
2	3	0.7	0.5	0.5
3	3	0.8	0.9	0.1
3	4	0.2	0.5	0.5

An arbitrary sequence of observations, e.g., 'HTH', can be generated by this model: the following state sequences all generate 'HTH' from initial state 1: 1222, 1223, 1233, 1234, 1333, 1334. The probability of the generated 'HTH' sequence varies, though, according to the state sequence used. Note, too, that of the possible state sequences, only two end in the final state (4). The probabilities for the sequences are computed as the products of the transition probability and the output symbol probability for each transition along the path; this yields:

State Sequence	Path Probability	Overall Probability
1222	(0.6)(0.5) (0.3)(0.8) (0.3)(0.2)	0.0043
1223	(0.6)(0.5) (0.3)(0.8) (0.7)(0.5)	0.0252
1233	(0.6)(0.5) (0.7)(0.5) (0.8)(0.9)	0.0756
1234	(0.6)(0.5) (0.7)(0.5) (0.2)(0.5)	0.0105
1333	(0.4)(0.9) (0.8)(0.1) (0.8)(0.9)	0.0207
1334	(0.4)(0.9) (0.8)(0.1) (0.2)(0.5)	0.0029

From this it is clear that the best state sequence, i.e., the one having the highest probability, is 1233. It is also clear that the probability with which this model generates the sequence 'HTH' is 0.1392, the sum of the probabilities of each path capable of generating the observation. This probability is referred to as the HMM's *score* for

the particular observation sequence†.

The parameters (transition probabilities, emission probabilities) of this model determine that a particular sequence of 'H' and 'T' is more probable than another. The same topology of model, with different parameters, might provide a high probability for generating a different string of 'H' and 'T.' It should be apparent that a set of these models could also be used to provide a probability measure for how likely an arbitrary sequence of 'H' and 'T' is. For example, suppose there are several HMMs like that in Figure 3. Each has different probability values associated with transitions and symbol generation. One model's parameters may lead it to generate a sequence of 'H' with very high probability, and any other sequence with very low probability. Another may similarly favour sequences of 'T', another alternating 'H' and 'T', etc.

Suppose we have an observation consisting of an arbitrary sequence of 'H' and 'T.' Each HMM can generate this observed sequence and provide the probability with which the model generates that sequence. Suppose that the observation sequence consists only of repeated 'H'. If we look at the scores from each of the HMMs, there should be one that is distinguishably higher, that provided by the HMM whose parameters lead it to generate a repeating sequence of 'H' with highest probability. The probabilities reported by all of the other HMMs will be much lower. If we adopt the policy of selecting from the scores reported by the different HMMs only the highest score, then we are implicitly selecting one of the HMMs from the set. One might, then, say that the observed sequence has been 'recognized' as being a repeating sequence of 'H' by one of the HMMs. Of course, this assumes that the probabilities involved have (somehow) been correctly assigned within each model.

We can summarize the process of recognizing sequences of 'H' and 'T' with HMMs as follows. Let the observation be $\mathbf{Y} = y_1, y_2, \cdots, y_T$, and assume we have a set of

† This score is that of the overall model, and is not to be confused with the score of an individual path. A path's score may be reported at any state q_i as the accumulated products of (transition × emission probabilities). For example, the score of the first path shown above (state sequence 1222) after the first transition is (0.6)(0.5), after the second (0.6)(0.5) × (0.3)(0.8), etc.

HMMs, $Z = \{H_1, H_2, \cdots, H_k\}$. We want to find the model $H_j \in Z$ for which $P(\mathbf{Y} \mid H_j)$, $1 \le j \le k$ has the highest value. We will then claim that H_j has 'recognized' the observation $\mathbf{Y}$. To compute these probabilities, we begin with the initial state q_0 of H_1. For each transition from this state a search *path* is begun. In the case of the HMM of Figure 3, two paths would be begun: one from q_1 to q_2, the other from q_1 to q_3. Each path will have a score which is the product of the transition probability × probability of emitting the observation symbol y_1. Now, for each current path, take transitions from the current state to states reachable from that state. In Figure 3, for the path now 'at' q_2, extend that path to include a next state q_3. The path ending at q_2 may also be extended to new state q_2. Similarly, the path ending at q_3 may be extended to new state q_4; it may also be extended to new state q_3. Thus, there are now four individual paths being explored: $q_1q_2q_3$, $q_1q_2q_2$, $q_1q_3q_4$, and, $q_1q_3q_3$. For the most recent expansion step, the scores of the paths are updated to be the current score × (probability of the chosen transition × probability of emitting y_2 on the particular transition).

This procedure continues, advancing one state at a time, 'reading' one input symbol (y_i) at a time, until the input observation string is exhausted, and HMM H_1 has produced an output string of length T for each path explored, and from these the model's score is computed. The procedure is said to be 'time synchronous' because each forward step in the search occurs by consuming one input symbol, which itself corresponds to some discrete event (e.g., a coin toss) or a discrete amount of time in the interval $1..T$.

Once this exercise has been performed over all Z, the $H_j \in Z$ with the highest score is said to be the model 'recognizing' the observation $\mathbf{Y}$.

For an HMM modeling a phoneme in a speech recognizer, the model is similarly time synchronously driven. Rather than a sequence of 'H' and 'T' observation symbols, the HMMs 'see', as an observation, a sequence of feature vectors. Each symbol y_i in the observation $\mathbf{Y} = y_1, y_2, \cdots, y_T$, is a u-dimensional vector, where u is the number

of features computed by the feature extractor per frame of speech (see § 2.2.2). The great power behind the success of HMMs as speech recognizers lies in their ability to model the two critical sources of variation in speech: duration and spectral energy. The probabilities associated with the state transitions in the HMM model the time (durational) variability of a phoneme, and the symbol emission probabilities model the spectral variability.

2.2.4 Hidden Markov Models

Formally, a hidden Markov model is a finite automaton generating output:

$$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

where:

Q is a finite number of states q_i,

Σ is a finite input alphabet of symbols σ_i,

Δ is a finite output alphabet, here chosen to be the same as Σ,

δ is a transition function performing a mapping $Q \times \Sigma \rightarrow Q$, i.e., $\delta(q_i, \sigma_k) = q_j$; $q_i, q_j, \in Q$ for any σ_k in Σ,

λ is a mapping function from $Q \times \Sigma \rightarrow \Delta$, i.e., $\lambda(q_i, \sigma_k) \rightarrow$ output symbol associated with transition from state q_i on input symbol σ_k,

q_0 is an initial state chosen from a set of initial states, $\pi = \{\pi_i\}$, $\pi_i = P(q_0 = i)$, $1 \le i \le n$, where $n = |Q|$

In general, an output generating finite state automaton may be characterized as either a Mealy machine or a Moore machine[24]; here we adopt the former view, as the definition above reflects‡.

Two fundamental assumptions pertinent to an HMM are:

(1) the probability of being in a particular state at time $t+1$ depends only on the state at time t (Markovian assumption),

(2) the probability that a particular symbol is emitted at time t depends only on the transition taken at time t (output independence assumption).

Models used in speech recognition tend to be 'left to right' models, reflecting the time evolution of the process being modeled, so $i \rightarrow j$ transitions occur where $j \ge i$†.

The mappings δ and λ are based on probabilities which are parameters of the model. Typically these probabilities are described as:

a_{ij} the probability of taking the transition from q_i to q_j. From a given state, the probabilities across all outgoing transitions must sum to 1.

---

‡ The Moore machine outputs symbols upon arriving in a state, as opposed to the Mealy machine which outputs symbols upon taking a transition. The models are provably equivalent, but the definition of λ is different between the two.

† This assumes an implicit ordered labeling of the states in the model, from left-to-right (as seen in Figure 3). In the general case, there may be transitions from a state to *any* other state in an HMM. Due to the nature of the process being modeled (speech), the transitions between states are such that no state other than the current state is visited again. This leads to models that are depicted as a left-to-right string of states, with an intuitive ordered labeling.

$b_{ij}(\sigma)$ the probability of outputting symbol σ when taking the transition from q_i to q_j. For a given transition, the sum of the probabilities of outputting symbols must sum to 1.

For any transition, the probability of outputting each of the k possible symbols is drawn from a distribution $d(X)$. In cases where the observations to be modeled consist of discrete symbols (as, e.g., coin tossing, or, when acoustic features have been vector quantized†) the output distribution $d(X)$ provides probabilities for each of the K possible symbol values:

$$d(X) = P(X|\, d), X \in \{1, 2, \cdots, K\}.$$

When the observations consist of continuous-valued observations (e.g., u-dimensional vectors of real numbers from a feature extractor), continuous distributions are used. The most popular choice of distribution function is a multi-variate Gaussian:

$$d(X) = \frac{1}{|\Psi|^{1/2}\,(2\pi)^{m/2}}\, e^{-1/2\,(X-\mu)^T \Psi^{-1}(X-\mu)}$$

where

m is the dimensionality of observation vector X,

μ is the mean vector of distribution d,

Ψ is the covariance matrix of distribution d.

Since acoustic feature distributions are not unimodal, a single Gaussian does not model them well. More accurate modeling is possible using a weighted *mixture* of Gaussians [41]:

† Vector quantization is a way to represent values in a continuous space more economically as discrete values. One constructs a *codebook* containing some number of prototype vectors or *codewords*. Powers of two are popular sizes for codebooks; eight is often used, for a codebook of 256 codewords. The prototype vectors are distributed throughout the space of continuous-valued data. Each continuous-valued datum can then be represented by the index in the codebook of the codeword to which it is closest in the space. See [21] for a complete description.

$$P_{mix}(X) = \sum_{k=1}^{K} w_k P_k(\underline{X}) ,$$

where $\sum_{k=1}^{K} w_k = 1$.

The 'richness' of distributions need not be constant across an HMM, i.e., not all transitions need have the same number of distributions associated with them.

The transition and emission distributions for a phoneme modeling HMM cannot be precisely determined. They are, rather, estimated through the use of data and a training procedure (see § 2.2.7). In some cases it may be desirable to reduce the number of emission distributions, e.g., when adequate training data is unavailable. This can be done by sharing distributions between related transitions. Distributions so shared are said to be *tied*.

An HMM may also contain transitions which have no corresponding output distributions; such transitions are said to be *empty* (see, e.g., the 'loop transition' of Figure 5).

There are three classic problems related to HMMs and their use: evaluation, decoding and training. The evaluation problem is concerned with determining the probability with which an HMM generates an observation. The decoding problem concerns how to determine the optimal sequence of state transitions through a network of HMMs. Training of HMMs is concerned with how to use training data to derive estimates for the parameters of the model. These are discussed in the next three subsections.

2.2.5 Evaluation

For a moment, consider an HMM which models not a phoneme but a short word, say a single digit. Such a digit recognizer may be built (see Figure 4) by feeding the acoustic feature stream as input to each of the word models in parallel. At the end of the

acoustic input for the word, the model which has the highest score is picked as signaling the digit that was spoken. How is this score computed?

Let **A** be an acoustic observation $a_1, a_2, \cdots, a_t$ obtained from the uttering of one of the words in the digit vocabulary. From this *acoustic* observation a corresponding observation vector of features is derived (see § 2.2.2). Each of the models in the recognizer will generate a string of 'observation symbols' $\mathbf{Y} = y_1, y_2, \cdots, y_T$ with some probability. Assuming that the models have been well trained, one model should show a distinguishably higher probability of generating a string 'close' to **Y**.

The probability that a model generates a particular output string **Y** is obtained by summing the probabilities of all paths capable of generating such a string, i.e., all

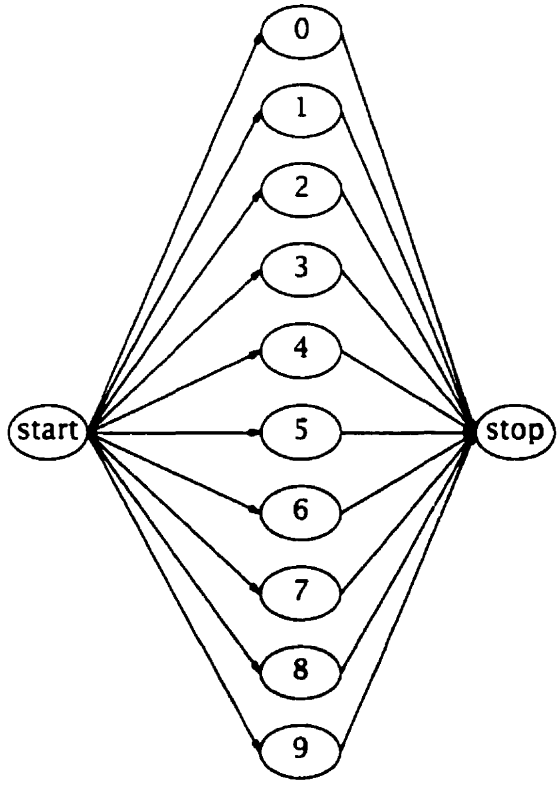

Figure 4 Isolated digit recognizer using HMM-based whole word models

paths having T transitions through the model. At a transition from state q_{t-1} to q_t along a state sequence $Q=q_0, q_1, q_2, \cdots, q_T$, symbol y_t is emitted as output. The probability of a particular sequence $\mathbf{Y}$ on a particular sequence of states Q under the output independence assumption is:

$$P(\mathbf{Y}|Q) = b_{0,1}(y_1)b_{1,2}(y_2)b_{2,3}(y_3)\cdots b_{T-1,T}(y_T)$$

$$= \prod_{t=1}^{T} b_{t-1,t}(y_t) .$$

The probability of the state sequence Q itself is given by

$$P(Q) = a_{0,1}a_{1,2}a_{2,3}\cdots a_{T-1,T}$$

$$= \prod_{t=1}^{T} a_{t-1,t} .$$

Combining these gives the probability that the model produces the observation sequence $\mathbf{Y}$ using the particular state sequence Q under consideration:

$$P(\mathbf{Y}, Q) = \prod_{t=1}^{T} a_{t-1,t}b_{t-1,t}(y_t) .$$

But any state sequence having T transitions can generate the observation sequence (with some probability), so it is necessary to sum over all such state sequences through the model. Let Q_T be the set of all state sequences having T transitions for the model. Then:

$$P(\mathbf{Y}) = \sum_{Q \in Q_T} P(\mathbf{Y}, Q) .$$

$$= \sum_{Q \in Q_T} \prod_{t=1}^{T} a_{t-1,t}b_{t-1,t}(y_t)$$

This is the probability that the model generated the observation $\mathbf{Y}$, i.e., the score reported for the model and used in choosing the best scoring model. Unfortunately, the fanout resulting from enumerating all paths of length T is exponential, $2TN^T$, N

being the number of states reachable from a current state, making direct evaluation of this product prohibitively expensive.

Fortunately, a more cost effective, recursive, technique is available: the forward-backward algorithm [5]. Let $\alpha_t(j) = P(y_1, y_2, \cdots, y_t, q_j)$, the probability of the partial observation string $y_1, y_2, \cdots, y_t$, and being in state j at time t. Let

$$\alpha_t(j) = \begin{cases} 0 & \text{if } t = 0 \text{ and } q_j \text{ not an initial state} \\ 1 & \text{if } t = 0 \text{ and } q_j \text{ is an initial state} \\ \sum_i \alpha_{t-1}(i) a_{ij} b_{ij}(y_t), t > 0 & \text{otherwise .} \end{cases}$$

The idea here is that in order to be in state q_j at time t, one need only consider each of the possible states from which q_j may be reached from time $t-1$. Being in some state q_i at $t-1$ implies having already generated some partial observation sequence ending with y_{t-1}; the probability of this situation is $\alpha_{t-1}(i)$. The probability of reaching state q_j at time t from state q_i at time $t-1$ is a_{ij}. The probability of generating the new observation symbol y_t on this transition is simply $b_{ij}(y_t)$. Thus the probability of arriving at state q_j at time t from state q_i at time $t-1$ with this partial observation string is $\alpha_{t-1}\ a_{ij}\ b_{ij}(y_t)$. This quantity is summed over all i reflecting the different q_i from which q_j is reachable in one time step.

The computation cost has now been reduced to a more reasonable N^2T.

This section began by presenting an application of the 'evaluation problem' of HMMs: determining the probability that a model generates a particular observation sequence. The forward-backward algorithm provides a feasible means of performing this computation. For the digit recognizer, this computation is performed on each of the models and the model for which the probability (score) is highest is hypothesized as the digit spoken.

2.2.6 Decoding

Consider an improvement to the previous section's digit recognizer — the ability to recognize a continuous stream of spoken digits of unknown length. The improvement is simple in principle: loop each digit's model back to an initial state (see Figure 5). Since an unknown number of iterations through this looped model will have occurred, knowing the one highest scoring model at the end of spoken input is not helpful. Recognition now requires knowing the best sequence of states through the digit models. By 'best' is meant the sequence of states providing the highest (global)

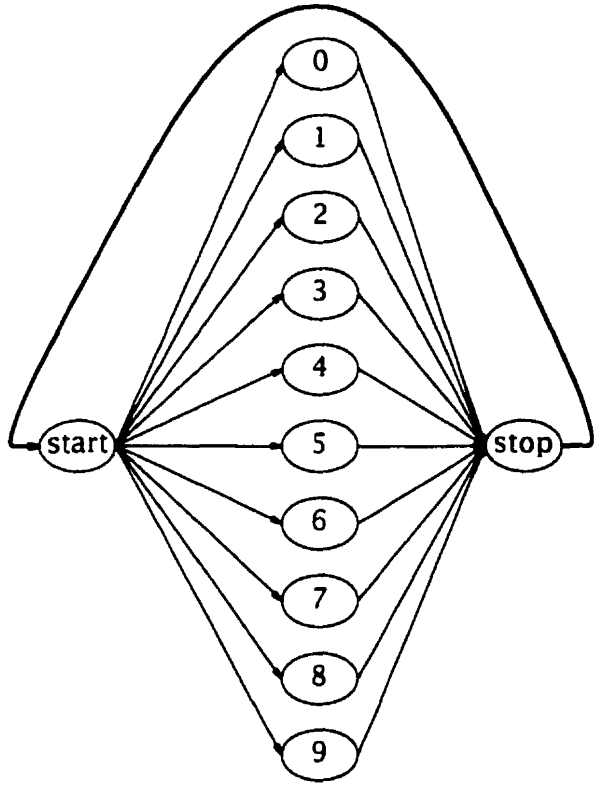

Figure 5 Continuous digit recognizer using HMM-based whole word models, looped to allow recognition of digit strings of arbitrary length.

score for the overall observation string, as opposed to, e.g., the sequence of individual (local) highest scoring states. A now popular algorithm for determining this sequence was introduced by Viterbi [63].

Let the globally best state sequence be $\mathbf{Q} = (q_0, q_2, \cdots, q_T)$ corresponding to the observation sequence $\mathbf{Y} = (y_1, y_2, \cdots, y_T)$. Then

$$\delta_t(i) = \max_{q_0, q_1, \cdots, q_{t-1}} P(q_0 q_1 \cdots q_{t-1}, q_t = i, y_1 y_2 \cdots y_t)$$

is, at time t, the path with the highest probability generating the first t observations and ending in state i. It is then the case that

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_{ij}(y_{t+1})$$

If, at each time t and for each j, the argument which maximizes the above is kept, it will be possible to backtrack through the kept arguments to hypothesize the highest probability state sequence.

The procedure as described above keeps track of *every* possible path 'explored' during recognition. In doing this, it assures that the globally optimal path will be identified, i.e., it is an *admissible* search procedure. In practice, many of these paths are wrong. As a way to prune the search space, a parameter is introduced which limits the paths (based on their scores-to-date) that will be accepted for further propagation. Use of this threshold, called a *beam*, renders the search inadmissible, but in practice, can result in very large savings in search time with little impact on recognition accuracy. If, early in exploring a path, the score is very low, it is very unlikely that the path is correct. The value of the beam threshold is determined manually, as a tradeoff between speed and accuracy, and tuned for a particular task.

2.2.7 Training

Previous sections have described how HMMs can be used to provide probability 'scores' for observations they model. Crucial to the model's success is having correct parameter values — transition and emission probabilities. For HMMs used as acoustic models, it is not possible to determine these parameters directly, so they must be estimated from training data.

Let us suppose that we wish to train a model for the vowel /ae/. Having decided on a model topology and feature set, the next step is to have training data ready. Typically this data consists of a set of spoken utterances where instances of individual phonemes have been (manually or automatically) labeled. Training now consists of presenting instances of the target observation to the model and *re-estimating* the model's parameters so as to increase the probability of the observation string being generated by the model. If the probability of observing $\mathbf{Y}$ from initial model M is $P(\mathbf{Y}|M)$, training aims to adjust M's parameters to produce M' such that $P(\mathbf{Y}|M') > P(\mathbf{Y}|M)$. Training iterates, replacing M with each improved M' until no further improvement is made.

With each iteration in the re-estimation, one is hoping (eventually) to produce optimal values of a_{ij} and $b_i(k)$ using (for the case of discrete HMMs [52]):

$$a'_{ij} = \frac{a_{ij}\dfrac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik}\dfrac{\partial P}{\partial a_{ik}}}$$

$$b'_{ij}(k) = \frac{b_{ij}(k)\dfrac{\partial P}{\partial b_{ij}(k)}}{\sum_{l=1}^{M} b_{ij}(l)\dfrac{\partial P}{\partial b_{ij}(l)}}$$

where $N = |Q|$ and $M = |\Delta|$. These re-estimation formulae lead only to locally optimal parameter values, ultimately providing the maximum likelihood estimate (MLE) of the

HMM. MLE training is the most popular method in current use; others include maximum mutual information estimation [3], and minimum discrimination information [17]. Gradient descent techniques like simulated annealing [45] can also be used.

A persistent difficulty in training models is the availability of data which adequately covers the range of values to which the recognizer may be exposed. A speech corpus collected in the state of Alabama cannot be expected to present sample observations corresponding to speech from Cork County, Ireland. The TIMIT [18,31] corpus has come into near ubiquitous use for training models for U. S. English because it contains speech data collected from eight dialectically distinct regions of the U. S., and contains sentences which were designed to be 'phonetically balanced', i.e., to ensure a reasonable number of occurrences of all phonemes.

A huge amount of the research work in speech recognition is devoted to various aspects of model training: corpus collection, labeling, and training algorithms. While models and modeling benefit from this effort, the research work described in this thesis is, in large part, not overly affected by model quality. To be sure, as model quality varies, a recognizer (with or without multiple pronunciations) will perform better or more poorly. However, the argument has already been made that even if acoustic models were perfect, recognition errors would still occur. The use of multiple pronunciations can mitigate these types of misrecognitions. Further, since models are imperfect, some misrecognitions attributable to weaknesses of models can be corrected using multiple pronunciations. The cost of a correction at the PD level may be much lower than the alternative: retraining of models using the re-estimation procedures described above.

2.3 Language Model

The AM is able to provide a score of how well the observed acoustics match words. Since this match is based solely on acoustic constraints, it may not yield a

syntactically or semantically meaningful string of words. It is desirable, therefore, to require the recognizer's hypothesized word string to be subject not only to acoustic constraints, but also to constraints inherent in the language of the task. A *language model* supplies such a set of constraints on words and the sequences in which they may appear.

Let L be a language whose spoken utterances are to be recognized. A generative grammar, G, where $L = L(G)$, is described as:

$$G = (V, T, P, S)$$

where:

V is a finite set of variables,

T is a finite set of terminals,

P is a finite set of productions,

S is a start variable.

Productions in grammar G are of the form $\alpha \rightarrow \beta$ where $\alpha \in (V \cup T)^+$ and where $\beta \in (V \cup T)^*$. G is generative in that from the start variable S, $S \in V$, repeated application of productions generates all strings of terminals possible in L. If all the productions in G have the property that α contains no terminals, i.e., $\alpha \in V^+$, G is said to be *context free* since a production may be applied to any α without regard to the context in which α appears.

Unfortunately, such formal grammars are of limited utility for most natural language processing (NLP) tasks. This is mainly due to the extreme difficulty in creating a grammar which has good *coverage*, i.e., can generate a suitably large variety of sentences in the natural language L_{NL}, and, at the same time, not *over-generate*, yielding sentences not in L_{NL}. One or the other of these is relatively easy to accomplish, but not both simultaneously. An example of an over-generating grammar is G_{WP}, generating all pairs of words in the vocabulary of L_{NL}.

The attraction of an over-generating grammar lies in its breadth of coverage of L_{NL}. By associating probabilities to the productions in G, it is possible to provide probabilities for each generated string of terminals so that those strings not truly in L_{NL} have low probabilities compared to those that are in L_{NL}. In this way the coverage provided by G is exploited profitably while the impact of G's over-generating is reduced.

Some of these grammars may be utilized effectively in speech recognition by representing them as stochastic finite state automata (SFSA). In so representing a grammar, states of the SFSA correspond to collections of variables and transitions are labeled with terminals; see Figure 6. For each transition designated by a terminal, one may substitute an automaton which realizes the terminal. In the applications of interest here, each terminal (a word in a presumed sentence in L_{NL}) can be replaced by an automaton representing the time evolution of the terminal's pronunciation. This automaton, with each state corresponding to a phoneme, may then itself be replaced by a series of other automata, namely, the HMMS modeling the acoustics of the individual phonemes. The result is a SFSA which has sequences of phoneme models labeling transitions. Such an automaton enforces the word-choice constraints desired of a LM and contains, embedded, those sequences of acoustic models legal in the language. That is, the SFSA combines into one *integrated stochastic network* both the LM and AM components of the recognizer.

The introduction of multiple pronunciations per word in such a network is straightforward: one substitutes the single phoneme sequence for a network of pronunciations for the word, all in parallel. !mprongram. What is less straightforward is how probabilities associated with the transition to one or another of the pronunciations are determined. Initially, one must assume that each of the pronunciations is equiprobable. Refinements to individual pronunciation probabilities can be made using observed frequency-of-occurrence counts from a set of training data, as, e.g.,

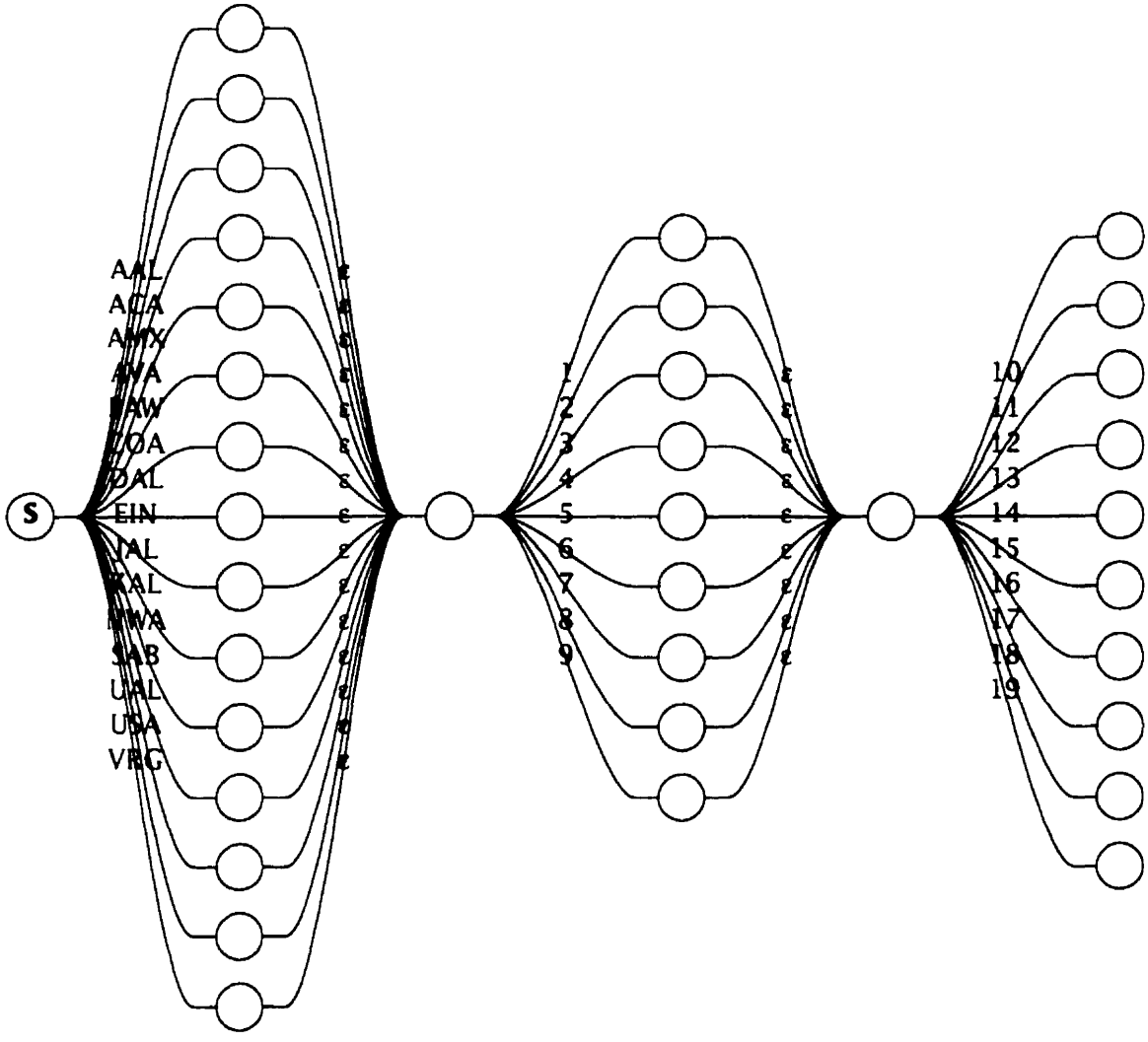

Figure 6 Segment of SFSA for airline callsigns Shown is the beginning portion of the SFSA, from 'S' start symbol through different airline names, then to segments of flight numbers, etc. The productions in the grammar which correspond to this segment resemble (terminals in italic):

S	→	call_sign command
call_sign	→	airline_name fl_num
airline_name	→	*AAL* | *ACA* | ... | *VRG*
fl_num	→	first second
first	→	*1* | *2* | *3* | ... | *9*
second	→	*10* | *11* | *12* | ... | *19*

[64]. More difficult is establishing context-sensitive pronunciation probabilities. The frequency-of-occurrence based technique reassigns transition probabilities to individual alternate pronunciations based on averages of observed numbers of occurrences over a training data set. But, in some cases, one pronunciation may be highly preferred over others, while in other cases it is not, e.g., /dh iy/ is a favoured pronunciation when the following word begins with a vowel, though in general, /dh ah/ may be observed to occur more often.

The introduction of more parallel branches within the network also has an unpleasant impact on the fanout of legal phoneme sequences.

In those cases where a rigid syntax is not practical, the SFSA based LM may be infeasible. For example, in natural language tasks users do not have to conform to anything more formal than the accepted base grammar for the natural language. As mentioned above, over generating grammars like G_{WP} are better suited to such cases because of their coverage, if suitable probabilities can be found to provide meaningful distinction between word pairs in L_{NL} and those not in L_{NL}.

One successful and widely used LM that aims to fulfill these requirements is that based on trigrams [26]. In the general case, the LM provides:

$$P(W_i \mid W_{i-1}, W_{i-2}, \cdots, W_{i-n})$$

i.e., the probability of word W_i given the n words which precede it. This is infeasible to compute in the general case; in practice, it is rare for n to be greater than 2 (trigrams); a bigram model uses $n=1$. The probabilities are estimated from counts of how often each m-gram appears in the training text.

A scarcity of training data often means that some trigrams may have zero or atypically low probabilities. One strategy, suggested by Jelinek, for dealing with this problem is the use of weighted frequency counts:

$$P(W_i \mid W_{i-1}, W_{i-2}) = q_3 \frac{C(W_{i-2}, W_{i-1}, W_i)}{C(W_{i-2}, W_{i-1})} + q_2 \frac{C(W_{i-1}, W_i)}{C(W_{i-1})} + q_1 C(W_i)$$

where $C(x)$ is a count of the number of occurrences of x, and, $q_3 + q_2 + q_1 = 1$.

In natural language tasks, e.g., WSJ, a bigram language model is a popular choice since it is easily trained and provides good coverage. In tasks where a more restrictive grammar can be used, e.g., ATS2, the option of an integrated stochastic network is both feasible and preferred.

One may have a choice of different LMs for a particular recognition task, e.g., bigram and trigram models may be available. How does one assess the quality of different language models for a task?

Consider a hypothesized word string $W_1, W_2, \cdots, W_n$ as generated by a language. If one adopts the view of a language as an information source with output symbols W_i, one can establish a measure of the *entropy* of the source as:

$$H = -\frac{1}{N} \log P(W_1, W_2, \cdots, W_N)$$

The information content of this source is that of some other source (language) which outputs words from a vocabulary of size 2^H with equal probability. Put differently, H is the (average) difficulty faced by a recognizer when it must determine a word from the source (task language).

In practice, only estimates of the probability of a word sequence are available, hence one may obtain the estimated entropy:

$$H_P = -\frac{1}{N} \log \hat{P}(W_1, W_2, \cdots, W_N)$$

The particular estimate $\hat{P}$, and hence H_P, depends upon the particular LM adopted. As no estimated word sequence probability will be better than the true (and unknowable) one, no estimated entropy H_P will be better than H. That (i.e., $H_P \geq H$) being the

case, a LM which results in an H_P closer to H is judged to be a better LM than one yielding a higher H_P.

It is common to assess LM performance using these notions, but expressing it as a metric called *perplexity*:

$$PP = 2^{H_P} = \hat{P}(W_1, W_2, \cdots, W_N)^{-\frac{1}{N}}.$$

The difficulty of the recognition task, using a particular LM that gave rise to $\hat{P}$, is that of recognizing strings originating from some other language (source) having PP equiprobable words. The perplexity of a task may thus be viewed as the average number of possible word choices from a given point.

3. Multiple Pronunciations and Belief

The core issue addressed in this research work is how to improve accuracy and reliability of recognition by endowing a recognizer's PD with a judicious selection of multiple pronunciations. For example, in the framework of the finite state grammars described above, each variant pronunciation is added in parallel with the canonical pronunciation so as to offer a path through the particular word which scores highly enough to keep the path under consideration and so prevent a competing and incorrect path from overtaking the correct one.

While it is true that the area of multiple pronunciations is not well explored, it is not *un*explored. This chapter briefly examines work done in this area, and shows how this research work is distinct.

One early effort developed a procedure for discovering spelling-to-sound rules by predicting a word's pronunciation from its spelling and a sample utterance [35]. The idea was that a word's spelling is in some way indicative of its pronunciation, and the transformation may be modeled using a noisy channel model. In developing the rules, it is important to find the pronunciation β maximizing:

$$\frac{P(\beta \mid s)\, P(u \mid \beta)}{P(su)}$$

where s is the spelled form of the word, and u is an utterance; $P(u \mid \beta)$ is computed by a phoneme recognizer. This is, in essence, the belief metric introduced in the section 3.4. Lucassen and Mercer developed the parameters for their noisy channel model using pronunciations derived from on-line dictionaries and one utterance per word. They also made use of binary decision trees to guide the selection of the best pro-

nunciation given a context† in their of a noisy channel. The proposed work will use different means of determining the analog to β, w_{i_k}, i.e., deriving these from observations in cases of misrecognition.

More recent work at IBM [2] improved on the earlier work in a number of ways, e.g., better decision trees, a larger collection of pronunciations and use of multiple utterances. The objective remained determination of pronunciation from a small number of utterances and a spelled form using spelling-to-sound rules.

3.1 Rule Driven Approaches

One might argue that phoneticians and linguists have, for many years, studied pronunciation, and that advantage could be taken of their observations; Table 3 shows some representative rules of this type. There are a number of impediments to using such observationally based rules. Firstly, while such rules are part of a body of published knowledge, they are not generally represented in ways that make them amenable to integration into automated recognizers. Their application can, however, be undertaken manually. Doing this by hand for a very small task may be reasonable, but, in general, manual construction of variants from rules is perilous. As Sloboda [57] points out, manual application of rules from one or more phoneticians opens one to the following problems:

(1) inconsistent use of phonetic units as number of words in PD grows [also as number of phoneticians grows],

† They define a *channel context* to be the "... current letter together with its literal [adjacent letters] and phonemic [phonemes associated with adjacent letters] contexts ..." It is a feature vector representation of this channel context for which the best pronunciation is sought. The decision trees are used to suggest which features to examine (and in which order) so as to arrive at a determination of the pronunciation π best suiting this particular channel context, while looking (preferably) at as few of the individual features as possible.

(2) the phonetically plausible pronunciation recommended by a phonetician may not be the most frequent or most likely pronunciation actually encountered,

(3) the number and variety of multiple pronunciations used in spontaneous speech may be difficult to predict using only phonological rules,

(4) maintaining correct information about which pronunciations are frequently used (i.e., relevant), for all pronunciations, is hard for humans to do consistently and reliably.

Hence, one requires an automatic means of developing and applying these rules, implying that they must be suitably represented. Moreover, given points (2) and (4) above, a data-driven scheme may perform better for a particular recognition task than rules drawn from more general purpose observations of phoneticians. Further, any effect observed to occur often enough to result in generating and enforcing a rule should resemble one suggested by a phonetician or linguist.

Secondly, even if expert phonetician rules are expressed in some suitable machine representation, not all of the 'expertise' lies in the rule itself. There is also the question of when to apply a particular rule, as not all rules are 'obligatory', i.e., apply in all cases. In addition to the rule itself, one wants to have some quantitative notion of how applicable a rule is, to guide when one might use it.

Other rules or sets of rules may be appropriate or inappropriate depending on, e.g., the dialect of the speaker. Recall the substitution 'rule' ae r → eh r as described in § 2.1.3: this rule would have a high probability of being applicable for a native English speaker from the United States west coast, but not from the northeast [1]. Any approach which aspires to speaker independence should be able to accommodate such dynamic shifts in rules or rule sets in preference to aiming for rules so broad in their coverage that they fail to provide adequate discriminability.

Many of the true speech-related misrecognitions are identifiable using *optional* phonological rules. For example, one might have a rule for reduction of a syllabic /n/ of form

/[ax ix] n/ → /en/

Such rules explain how a particular variant pronunciation may occur, but are optional in that the speaker may or may not always follow the rule, resulting in one or another pronunciation of the word. Thus, having a set of optional rules tells one that a particular pronunciation *may* occur instead of the canonical pronunciation (or another variant), but does not provide any measure of the likelihood of a particular variant pronunciation occurring. One solution is to attribute to each rule a probability of it being applied.

Given a set of optional phonological rules, Tajchman, Jurafsky and Fosler [61] describe a technique for learning the probabilities of such rules (see Table 3). Beginning with the rules whose probabilities are to be determined, they apply each of the rules to a large merged lexicon (> 75 K words) to generate an expanded lexicon of surface forms. This latter lexicon is then used in recognition experiments on a large corpus (WSJ0) to establish counts of how often each surface form in the expanded lexicon occurred. From these counts, and knowing which of the optional rules led to the particular surface form, count-based probabilities for the individual rules were

Table 3 Optional rules whose probabilities are determined in [61]

Type	Rule
Reduction: mid vowels	-stress [aa ae ah ao eh er ey ow uh] → ax
Reduction: high vowels	-stress [iy ih uw] → ix
Reduction: R-vowel	-stress er → axr
Reduction: syllabic n	[ax ix] n → en
Reduction: syllabic m	[ax ix] m → em
Reduction: syllabic l	[ax ix] l → el
Reduction: syllabic r	[ax ix] r → axr
Flap:	[tcl dcl][t d] → dx /V ... [ax ix axr]
Flap r:	[tcl dcl][t d] → dx /V r ... [ax ix axr]
H-voicing:	hh → hv / [+ voiced] ... [+ voiced]

established.

The technique must be given the rules *a priori*, it does not discover or infer them from data. The rules, once assigned probabilities, can be used to generate variant pronunciations such that each pronunciation has an associated probability. They conducted two tests on the female speaker subset of the WSJ task (1993 development set; 5,000 word vocabulary). In the first, which compared a PD with multiple pronunciations having all words equiprobable against one in which the pronunciation probabilities had been set based on rule probabilities, the observed 6.7% reduction in word error rate was claimed to be statistically insignificant. A second test used a PD in which pronunciation probabilities were computed based on rule probabilities and pronunciations having sub-threshold probabilities were removed. Depending on the value of the threshold, a 16% to 20% reduction in word error rate was reported. Smaller PDs which outperform larger ones are attractive.

Other work does aim to discover rules from actual speech data. A speaker dependent system developed for Japanese [25] uses training speech from the target speaker, S, accompanied by a phonetic transcription, P, containing only canonical pronunciations for the utterances in S. Viterbi alignment of sentences in P to the corresponding recognized phoneme string provides the likelihood $L(S_k \mid P_k)$ for sentence k. In addition, individual phoneme likelihoods and durations for each phoneme m (average μ_m, standard deviation σ_m and minimum duration τ_m) are determined from the (HMM) models used for recognition. These models are trained on a large corpus of speakers, i.e., are not specific to the speakers for whom rules are sought. Tentative phonological rules are deduced from the recognizer output in the categories of insertion, deletion and substitution (of 1 or 2 consecutive phonemes).

The notion for a tentative rule is, for the case of deletion, that if / a c / is recognized for / a b c /, then the acoustic model for the deleted / b / should show either a shorter duration or lower likelihood than the canonical case would. If, in sentence k, x_i is the i^{th} phoneme in canonical phoneme string P_k, an instance of phoneme m,

having duration t_i, and one of

$$t_i < \mu_m - 2\sigma_m$$

or $t_i < \tau_m$

or $L(x_i) < L(m)$

is true, then there are grounds to support generation of P'_k which contains the deletion. A new tentative rule is then introduced if $L(S_k \mid P'_k) > L(S_k \mid P_k)$.

Substitution and insertion rules are analogously introduced. Each tentative rule is then tested on the entire set of training sentences. Two criteria are applied to the result of this test. The tentative rule becomes a 'full-fledged' rule if (1) the likelihood scores for other speech that matches the left hand side of the rule are improved by use of the rule, or, (2) use of rule-defined multiple pronunciations applied to training speech sentences improves discriminability. If a tentative rule fails both of these conditions, it is deleted.

The authors report an average improvement of 2.4% in recognition using a multiple pronunciation dictionary constructed using rules they inferred (in one case, 1,026 tentative rules, reduced to 599 'real' rules). A weakness of this approach, though, is that no rules can be inferred for contexts not encountered in the training data. Also, no generalization is made from the rules that are retained to produce a smaller sized, more general set of rules. Lastly, inferred rules are speaker dependent.

3.2 Non-Rule Driven Approaches

The point in having rules is to have a constrained means of generating multiple pronunciations. But use of rules, either discovered or assigned *a priori* (without probability), is not the only way to accomplish this. Taking the view that multiple

pronunciations are the result of a learnable mapping between a canonical form and a particular realization of that form, i.e., a phoneme → phone mapping, Riley [53] describes a technique for using classification trees [8] to predict phoneme realization (in context). The trees are grown on a context labeled set of over 100,000 phonemes drawn from TIMIT's si and sx sentences. On a reserved independent set of 336 si and sx test sentences, the tree predicted correct phonemic realizations 84.1% of the time. Handling of insertions is largely special-cased in this approach, and for TIMIT where the 37 most common insertions account for 95% of all insertions, this turns out not to be unreasonable. The resulting prediction rate is slightly reduced, to 83.3%, when trees are grown containing insertion pairs (the phoneme and associated insertion treated as a pair).

Riley's approach is interesting also in that no speech recognition was used to develop the trees. Rather, Bell Lab's text-to-speech system [13] was used to generate phonetic transcriptions from the text transcriptions of the TIMIT sentences, and these compared against the phonetic transcriptions in TIMIT. One supposes that if a recognition system were used that the trees would acquire some predictive power based on idiosyncrasies of the recognizer, in addition to that based on genuine speech events.

Another automatic approach for building multiple pronunciations directly is described by Wooters [64]. His approach begins with, for each word, a model which is the concatenation of phoneme models representing a single (canonical) pronunciation. Then, using a PD derived from combining several sources, including TIMIT, providing 160,000 words with 300,000 pronunciations [61], pronunciation models are augmented by adding a new path (in parallel) for each novel pronunciation encountered. To the resulting augmented model containing multiple pronunciations of the word is applied an adaptation step which adjusts probabilities within the model to reflect training data more realistically. Following adaptation, the pronunciation probabilities are re-estimated using an algorithm proposed by Stolcke and

Omohundro [59]. In this step, the probabilities are not only re-estimated, but the model structure changed by merging of states so as to yield a model more general than its predecessor. These two sequential steps can be iterated to produce models which better fit the recognition task. The result is a smaller model which may even be able to model novel pronunciations, by virtue of paths which do not correspond to variants observed in the training data.

Tests of this approach, performed in the BeRP system [28], showed a 21% reduction in word error rate using a multiple pronunciation PD over a single pronunciation version.

More recent work also aimed to build alternate pronunciations from a corpus of real speech [10]. The authors report investigating the use of classification trees trained on data from the Switchboard and TIMIT tasks. A new dictionary was constructed containing canonical entries plus those obtained from application of the trees (one tree per phoneme). Trees were trained on both hand labeled and a few different automatically transcribed data sets. Use of the dictionaries so generated yielded a reduction in word error rate of 0.9% from a baseline reference of 44.7%.

Use of the dictionary generated with the trees trained on the hand labeled data resulted in a higher WER, leading to the authors' suggestion "...either the ... trees generalize incorrectly or do a poor job of assigning costs to the alternate pronunciations, which is crucial to the success of dictionary enhancement methods. We therefore examined a more conservative approach to dictionary enhancement."

That more conservative approach involved the construction of dictionaries from each of the hand labeled and automatically transcribed data sets used in the previous effort, but keeping only the most frequently observed pronunciations, which included some 'multiword' words to deal with between word coarticulation. Once again, the best WER reduction was 0.9%.

A WER reduction of 2.2% was reported following the retraining of acoustic models.

3.3 Dialect and Accent Work

Speech recognition systems need to be robust to wide variations of dialect not only in native English speakers, but also in speakers for whom English is not a mother tongue. Multiple pronunciations play a natural role in providing this robustness, by contributing pronunciations peculiar to the different non-native English speakers' dialects. Increasing the number of variant pronunciations per word is not without expense or risk: larger PDs are more time- and space-consuming, and increasing numbers of variants increase the chance for error arising from confusability of variants for different words. Ideally, then, one might wish to have a 'standard' PD, featuring a modest number of multiple pronunciations for 'standard' dialects, while having the ability to augment, dynamically, this PD with pronunciations found to increase accuracy when speakers of a particular dialect use the system. Such an ability requires not only the gathering and classification of such pronunciations, but, also, the ability to make a reliable determination of a speaker's dialect.

Work on the BeRP system includes handling of accents and dialects. Having shown that a system trained on American English experiences "...significantly more errors with non-native accents" [29], the focus of support for dialects is identification and subsequent modeling of the dialect. The authors report interesting results on identification, using either acoustic features alone (a multi-layer perceptron [MLP] network with one output unit per accent), or, using an analysis of sentence syntax†.

† BeRP is a natural language task, i.e., with no imposed sentence structure. The authors studied syntactic constructs which proved highly discriminative of accent. For example, American English speakers were found to be twice as likely as German English speakers to end a query with 'please' whereas the latter group was found to be 200 times more likely to begin a query with 'please.'

Once reliable determination of a speaker's accent or dialect is made, the recognizer can pick appropriate pronunciation models. Use of either technique produced roughly equivalent success, 62% for acoustic and 60% for syntax, at correct identification of speaker accent (at the sentence level). Use of a confidence measure and threshold for rejection of ambiguous sentences improved the rate at the 50% rejection level to 70% acoustic, 68% syntax, or, 73% combined.

Whereas BeRP can afford to rely on syntax, tasks where speakers have constrained syntax (e.g., ATS2) need to rely more heavily on acoustic means of distinguishing accent. Other recent work on the use of acoustics alone to identify a speaker's dialect does better than the abovementioned MLP, providing a (poorest) accuracy of 81.5% distinguishing between 'neutral', German, Turkish and Chinese accented English [22].

3.4 Belief

One observes, in speech recognition, an acoustic stream originating from some speaker. By the time it reaches the observer, this acoustic stream may be considerably altered — by various forms of stationary and non-stationary noise, by patterns of constructive and destructive interference from reflections off walls, floors and ceilings, and from other sources of speech. It may even reach the listener indirectly, e.g,. by telephone, with further modifications introduced by the mediator of the indirection. Even though the acoustic stream may have originated from the speaker in a comparatively pure state and is received without adulteration, it may not have contained the phoneme sequences for words one would expect from using only canonical pronunciations.

One way in which this speech process may be modeled is shown in Figure 7. Each 'transition' between 'states' in this figure suggests that the originating state in some way influences the value of the destination state. For example, the choice to

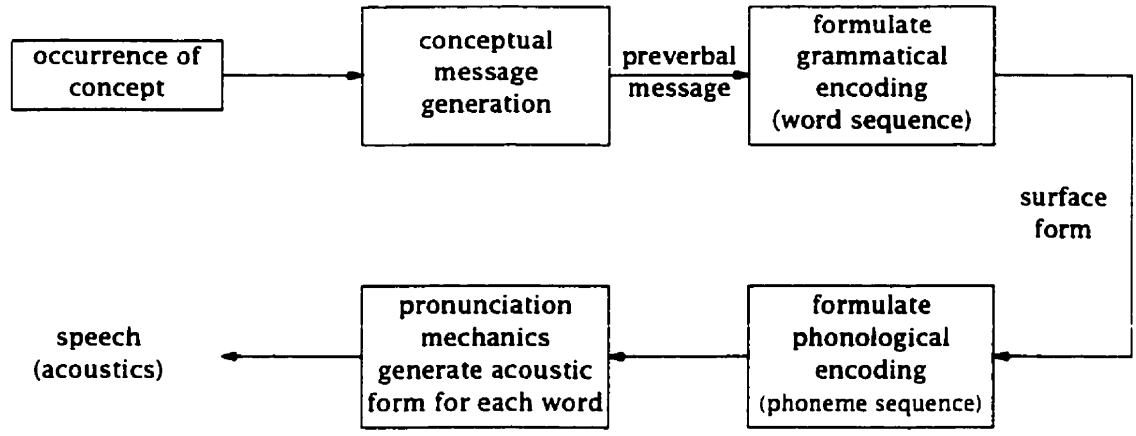

Figure 7 A causal network view of speech production from originating the concept to be communicated to the acoustic event of speech. The speaker's mental lexicon (not shown) is implicated at the two 'formulation' stages. A more complete characterization of the process would also include the speech recognition components the speaker itself uses to monitor its own speech (where, again, the lexicon is involved). The acoustic event 'speech' refers, minimally, to the spoken utterance as would occur in a quiet room in such a way that the hearer can perceive and recognize the speech. In reality, the 'room' may not be quiet (e.g., office background noise, a moving car), and the hearer may be remote to the speaker so that the speech is actually conveyed via telephone or radio, adding channel noise characteristic of the communication medium to the channel noise of the existing error-prone 'speech channel,' consisting of the last stages (bottom row) in the above causal network. (Figure adapted from Levelt [33]).

speak a particular word W_i directly influences the set of phonemes chosen which, in turn, affects the operation of the speech articulators. The result of this chain of influence is externally observable evidence, acoustics.

Of particular salience to this research work is the step in which some pronunciation w_{i_k} is selected for the chosen word W_i. In cases where w_{i_k} corresponds to the canonical pronunciation, w_i, a recognizer should readily succeed in recognizing the word. In cases where this is not true, it is expected to fail unless the recognizer has the ability to handle multiple pronunciations and has, among its multiple pronunciations, the variant chosen by the speaker. Examining this step may lead to a reliable means of providing the recognizer with the means of handling multiple pronunciations.

Figure 8 focuses attention on the particular step in the speech generating process concerned with selection of a specific pronunciation. For a fixed W_i there is one state representing the process by which a pronunciation w_{i_k} is selected. The consequence of this action is the generation of some sequence of acoustic events as the phoneme string comprising w_{i_k} is realized, with the result that externally observable evidence, **A**, is generated.

A formal model of this type of causal stream might provide a means of objectively assessing the relationship between the various w_{i_k} and w_i. Fortunately, just such a model exists: Pearl's *stochastic inference networks* (SIN) [46]; see Figure 9.

SINs have states characterized by variables (traditionally having upper case names) which can assume any one of the mutually exclusive values in the domain associated with that state. Connecting one state to another is a directed arc which conveys the concept that the value of the variable in the originating state influences

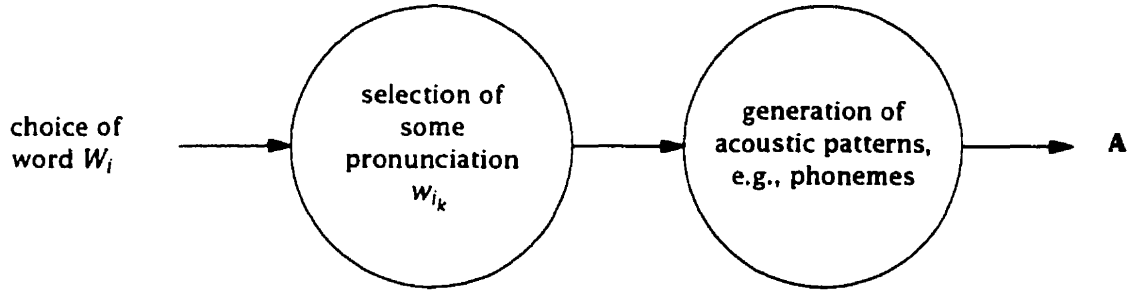

Figure 8 Selection of a pronunciation

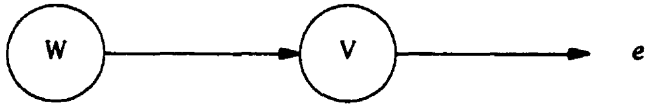

Figure 9 Simple stochastic inference network States are identified by random variables *W* and *V*; the directed edge captures the notion that the value of *W* from the domain of values it may take on in its state, influences the value *V* may take on in its state. *e* is externally visible evidence that *V* has a particular value.

the value of the variable in the destination state. In addition, a quantity called *belief* is defined; it conveys, as a probability, the belief in a variable having a particular value:

$$Bel(x) = P(x \mid e).$$

In the simple SIN of Figure 8, the $Bel(w_{i_k})$ is $Bel(w_{i_k}) = P(w_{i_k} \mid e)$ where the evidence, e, is observations from the acoustic stream, $\mathbf{Y}$, and the speaker's *intention* to utter W_i as w_i. That is, e represents the combined effect upon the target variable in the SIN of other instantiated variables able to influence it; in this case, those variables are $\mathbf{Y}$ and $W = W_i$. Thus,

$$Bel(w_{i_k}) = P(w_{i_k} \mid W_i \mathbf{Y})$$

$$= \frac{P(W_i \mathbf{Y} w_{i_k})}{P(W_i \mathbf{Y})}$$

$$= \frac{P(\mathbf{Y} \mid w_{i_k} W_i)\, P(w_{i_k} \mid W_i)\, P(W_i)}{P(W_i \mathbf{Y})}$$

Here it may be argued that the acoustic observation $\mathbf{Y}$ depends directly upon the phoneme string uttered which is exclusively a consequence of w_{i_k}, and not of W_i, hence $P(\mathbf{Y} \mid w_{i_k} W_i) = P(\mathbf{Y} \mid w_{i_k})$. Further, when considering different variant candidates of W_i, the denominator term $P(W_i \mathbf{Y})$ is constant, and may be ignored. This leaves

$$Bel(w_{i_k}) = \alpha\, P(\mathbf{Y} \mid w_{i_k})\, P(w_{i_k} \mid W_i)$$

If, in the case of a misrecognized word, one generates variant pronunciations according to some set of constraints, this belief can serve as a metric of how 'good' each generated variant is, and one may select the $n \geq 1$ variants having the highest belief scores for inclusion in the PD.

Since W_i is now allowed to be represented by one of its several variants, then:

$$P(W_i \mid \mathbf{Y}) = \sum_{k=1}^{K} P(w_{i_k} W_i \mid \mathbf{Y}) \propto \sum_{k=1}^{K} Bel(w_{i_k}).$$

A reasonable approximation to the above equality is:

$$P(W_i \mid \mathbf{Y}) \approx \max_k P(w_{i_k} W_i \mid \mathbf{Y})$$

In the above,

$$P(w_{i_k} W_i \mid \mathbf{Y}) = \frac{P(\mathbf{Y} W_i w_{i_k})}{P(\mathbf{Y})}$$

Thus, the probability that W_i can be hypothesized through w_{i_k} may be expressed as

$$\max_k \frac{P(\mathbf{Y} \mid w_{i_k})\, P(w_{i_k} \mid W_i)\, P(W_i)}{P(\mathbf{Y})} = \max_k Bel(w_{i_k})$$

For a given word, $\frac{P(W_i)}{P(\mathbf{Y})}$ is constant across all variants of W_i, so the above may be reduced to:

$$\max_k P(\mathbf{Y} \mid w_{i_k})\, P(w_{i_k} \mid W_i)$$

which is the maximization of the same quantity as in [35].

The first of these two probabilities is provided by the acoustic modeling component of the recognizer.

Given an observation $\mathbf{Y}$ of speech and a word W_i, finding the most likely pronunciation w_{i_k} implies knowing all K pronunciations of W_i. In general, these cannot be known, e.g., data sparseness is an inescapable problem, and consequently, $P(w_{i_k} \mid W_i)$ cannot be directly estimated from data. There are, fortunately, many ways of generating candidates w_{i_k} and there are many models which can be considered for computing $P(w_{i_k} \mid W_i)$.

Making the simple assumption that any phoneme may be replaced with any other phoneme and that neither deletions nor insertions occur, w_{i_k} can be found by a Viterbi search. If $w_i = f_{i_1} f_{i_2} \cdots f_{i_j} \cdots f_{i_J}$ and P_{ijk} is the probability of phoneme $f_k, k = 1, 2, \cdots, K$ being a substitution of f_{i_j}, then the HMM having the structure shown in Figure 10 can be used to compute the last term in the Belief expression.

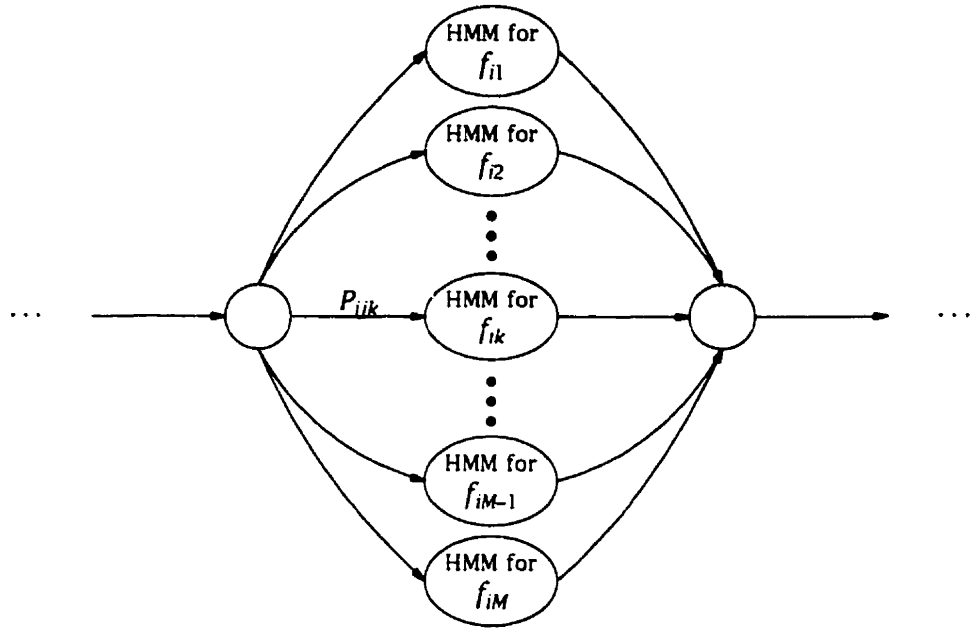

Figure 10 Simple HMM for deriving $P(w_{i_k})$ The structure shown in the figure is used at each phoneme position i in $P(w_i)$ to compute the probability that f_{ik} replaces f_{ij}, where j is in the canonical pronunciation at position i. The transition probabilities, P_{ijk}, reflect the probability of such a substitution.

This assumption is, however, too crude and more complex models have to be considered to generate all variant pronunciation candidates w_{i_m}. This is the case since, firstly, insertions and deletions with respect to the canonical pronunciation do occur, and, secondly, because of context-dependent substitutions. Thus a problem that emerges as very important is the strategy chosen for the generation of variant pronunciation candidates.

4. Making and Assessing Variant Pronunciations

Given the beneficial effect of multiple pronunciations, the critical question is how to determine *which* pronunciation variants to add to the PD.

An expert phonetician can look at the recognition errors that arise and generally provide insightful variant pronunciations that correct the error (see Table 1). The focus of this thesis work is to examine ways in which such expertise can be integrated into a recognition system at the level of the PD, and show that this is a viable means of improving recognizer accuracy. Rather than attempt to integrate an expert system in the domain of pronunciation, the hypothesis is made that inexpensive (compared to an expert system) rule-based or statistical (or hybrids of the two) generative methods can provide adequate performance improvements. This chapter looks at a variety of such inexpensive techniques.

Given that a pronunciation w_j has been identified as a misrecognition of word W_i, with canonical pronunciation w_i, and is indicated in the observation string by its starting and ending frame numbers, (t_b, t_e), the first step in correcting the misrecognition is to generate a set υ of variant pronunciation candidates (VPCs). The specific method used to generate this set is what distinguishes the techniques described here. Naturally, it is hoped that there is at least one $w_k \in \upsilon$ able to correct the misrecognition.

At issue are the composition and size of the set of variant pronunciation candidates. In terms of composition, one inclination is to use phonological rules, often determined by phoneticians, to guide construction of plausible multiple pronunciations [15]. Another possibility is to rely upon a data-driven approach in which the recognizer acquires variant pronunciations for words as it encounters and recovers from misrecognition.

As for |υ|, a large set is more likely to include a VPC that corrects the misrecognition than is a small set; moreover, a large set may contain several good candidates, allowing selection of the 'best' among the good. Unfortunately, in order to assess the merit of a candidate requires running recognition on part or all of the sentence again, with the VPC added to the PD. As this process is expensive, one wishes to perform it as few times as possible. Also, larger sets are more likely to contain pronunciations confusable with variant or canonical pronunciations of different words, increasing vulnerability to misrecognition. Hence, small sized sets of variant pronunciation candidates are desirable.

Nearly all of the experimental work described in this chapter was performed on the ATS2 task. Scores shown are log probabilities reported by the recognizer using a SFSA representation of the task grammar. The collection of recordings comprising the ATS2 set, some 84 sentences uttered by several speakers, was partitioned into 47 training sentences, 27 evaluation sentences and a test set of 10 sentences. For more details of the task and recognition system, see appendices 1 and 2.

4.1 Scoring

Irrespective of the method used to generate a VPC, a way of objectively assessing its value is needed. The belief score described in chapter 3 is central to the scoring. The objective is to introduce "successful" VPCs into the PD of a recognition system. As will be seen later, this needs to be done somewhat conservatively.

The initial procedure for evaluating a VPC begins by determining if the VPC succeeds in correcting the misrecognition in sentence context. That is, if the variant is introduced into the PD and recognition on the entire sentence is re-tried, is the misrecognized word now correctly recognized using the newly introduced candidate? If not, then this candidate can be dismissed from further consideration.

Candidates retained from this first step are ranked in order of their belief scores, $\alpha P(\mathbf{Y} \mid w_{i_k}) P(w_{i_k} \mid W_i)$. The value of α allows one to adjust the contribution of the two component (acoustic, pronunciation) probabilities. For all work reported here a fixed value of α was used.

The acoustic component of the belief score, returned by the recognizer itself, is the log probability of the best state sequence explaining the (acoustic) observations.

The pronunciation component of the belief score is determined using pronunciation performance models (see § 4.7). These models are trained on a corpus of labeled pronunciations of words – TIMIT in this case – to provide models analogous to the phoneme models used by the recognizer's acoustic matcher. These pronunciation models capture, for a given phoneme, how often it was observed as (i) itself, (ii) substituted by another phoneme, (iii) being deleted, (iv) having some other phoneme inserted before it (see Table 9).

A feature of the TIMIT corpus is that it contains labeled utterances collected in eight separate 'dialect' regions (see Appendix 1 for details). As such, pronunciation performance models can be made distinctive for individual regions capturing differences in pronunciation characteristics. Figure 12 illustrates the different pronunciation behaviour for /iy/ over the different dialectic regions. Figure 11 shows the effect of using the dialectic region specific probabilities upon the pronunciation component of the belief score. Note, in particular, the wide differences in the probabilities in the pronunciation of place names — one of the original motivations for this work. For nearly all work reported here, a single pronunciation performance model set, made by pooling the raw counts from all the individual dialectic regions, has been used.

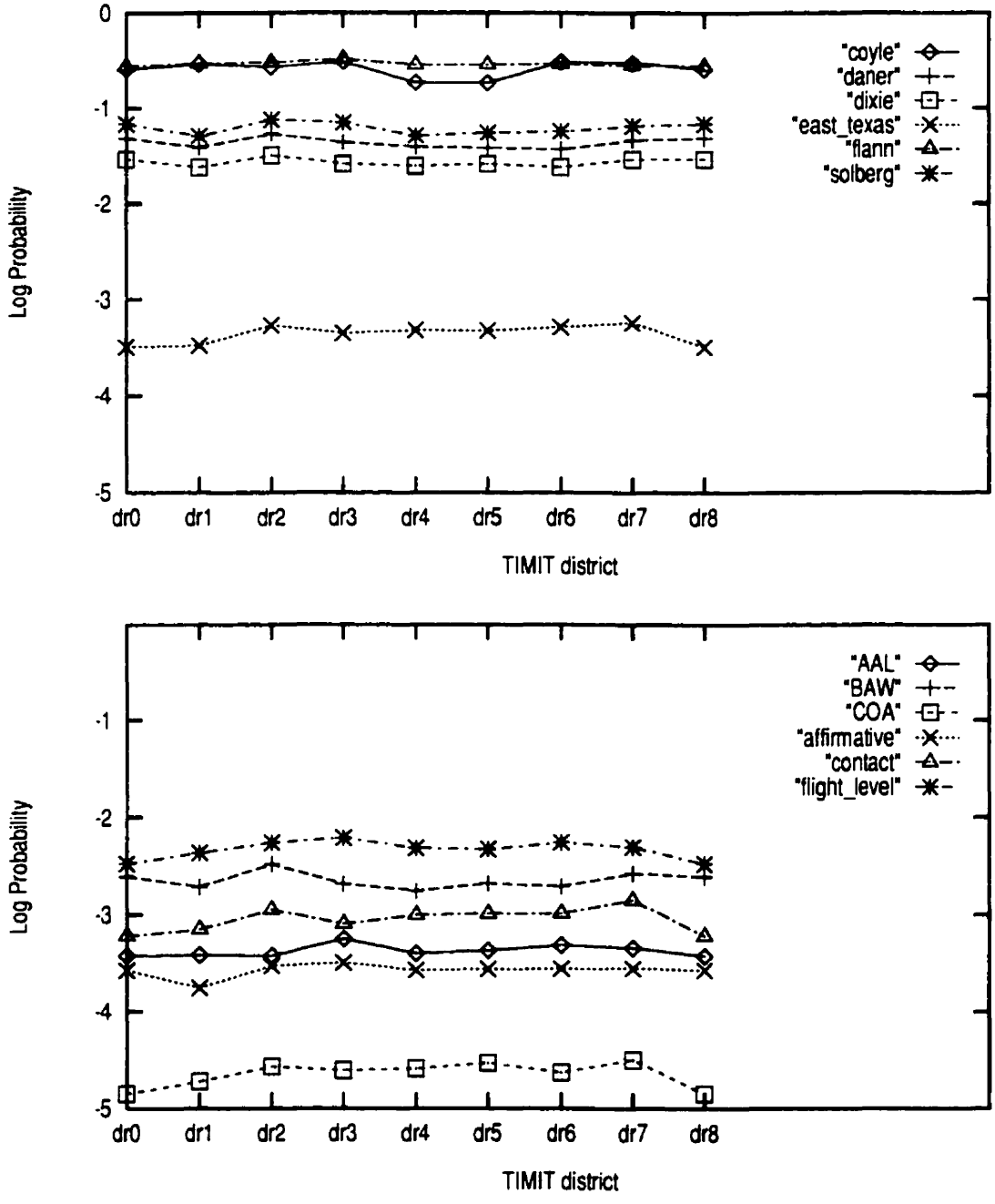

Figure 11 Pronunciation Probability Across Dialectic Regions Shown is the variation in $P(w_i \mid W_i)$ across the eight dialectic regions over which data were collected. The region 'dr0' is artificial and is the result of pooling scores from the eight districts dr1 - dr8. The upper plot shows some of the city names, the lower plot selected command words in the task.

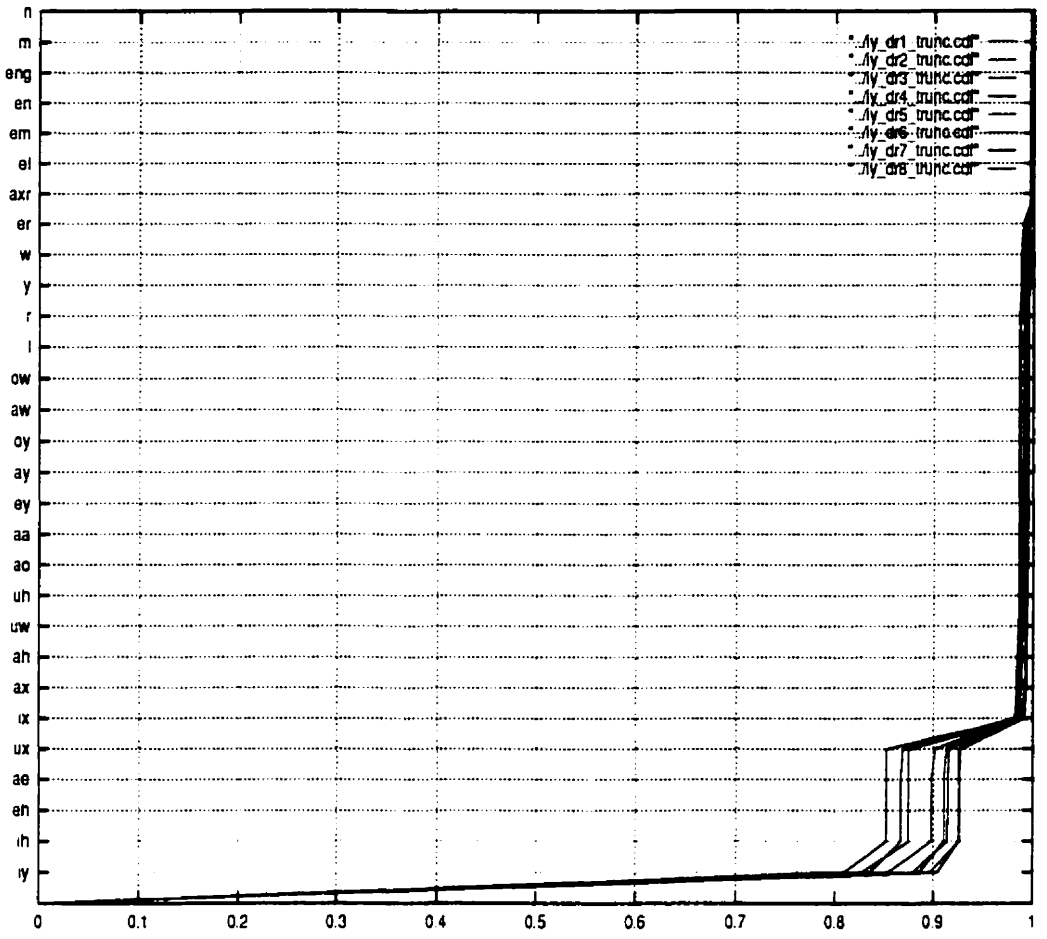

Figure 12 Regional Variations in Substitution and Deletion of a Phoneme The plot shows the cumulative distribution function for the probability of the phoneme /iy/ as substituted by the phonemes shown on the *y*-axis. While the largest contribution is, as expected, made by /iy/ itself, notable contributions are made by different vowels in the different dialectic regions. Note also that phonemes not in the class of vowels make no contribution to the cdf. Not all phonemes are shown; those not appearing on the *y* axis made no further contribution to the cdf.

4.2 An Obvious Suggestion

Suppose W_i has been misrecognized as W_j. The most obvious suggestion would be to add the observed phoneme string as a variant pronunciation of W_i to the PD. This is not a constructive strategy. Firstly, the hypothesized phoneme string often diverges significantly from the canonical forms for either of W_i and W_j. For example,

in the sentence "KAL 19 contact new york approach on one two zero point niner," (spoken by a Bangledeshi speaker of English), the word 'contact' is misrecognized as 'climb to.' The hypothesized phoneme string† where 'contact' should have been recognized is: gcl g aa dcl d ey kcl k. Secondly, the observed phoneme string, as a distortion of W_i, is highly speaker specific.

Adoption of this strategy might be suitable for a case where the vocabulary is small, containing words of low confusability, and one speaker will use the system. It is not suitable for medium or larger sized vocabularies, or where speaker independence is required.

Introduction of this pronunciation into the PD, while not recommended in practice, does perform one useful thing. The acoustic score of a recognition performed using this pronunciation for the misrecognized word is the best acoustic score a correct recognition of this sentence can have with the particular misrecognized word now correctly recognized. The acoustic score provided when the canonical score alone is used, and misrecognition resulted, is the worst score that may be reasonably expected (if one is attempting to obtain correct recognition). Thus, between the canonical pronunciation and the particular observed pronunciation, bounds are established for the value of acoustic score. These bounds are also the bounds (but opposite in 'polarity') for the belief component $P(w_{i_k} \mid W_i)$. That is, use of the best pronunciation (i.e., canonical) yields the worst acoustic score. The value of α may be used to vary the way belief values change within the range, but the bounds are fixed.

This case is typical, however, of misrecognitions in that the observed string is not obviously suggestive of 'contact' or 'climb to'. Apart from its utility in establishing bounds on belief scores, it should be evident that filling the PD with such 'far removed' and speaker specific alternative phoneme strings would not be fruitful.

† as determined by a pure phoneme recognition using a simple bigram language model, not trained on the ATS2 task (trained on TIMIT).

4.3 Iterative Transformation

Nor, as it turns out, is it necessary to add such 'far flung' pronunciations. If one begins with this hypothesized phoneme string and introduces variant pronunciations by transforming it, phoneme by phoneme, back towards the canonical form, one observes that the *only* critical difference is the last vowel (see Figure 13).

This suggests the following procedure:

(1) isolate phoneme string corresponding to misrecognized word,

(2) align misrecognized segment with canonical pronunciation of word,

(3) perform relaxation of observed pronunciation back towards canonical until some plausible variant w_r is found such that no pronunciation w_s is closer (e.g., in edit distance) to the canonical pronunciation and still able to correct the misrecognition.

Step 1 in the procedure is relatively straightforward if the correct version of the utterance is available since a rough segmentation of the hypothesized phoneme

canonical:	kcl	k	aa	n	tcl	t	ae	kcl	k	tcl	t
observed:	gcl	g	aa		dcl	d	ey	kcl	k		
var1:	gcl	g	aa		dcl	d	ey	kcl	k		
var2:	gcl	g	aa		**tcl**	**t**	ey	kcl	k		
var3:	gcl	g	aa		tcl	t	ey	kcl	k	**tcl**	**t**
var4:	**kcl**	**k**	aa		tcl	t	ey	kcl	k	tcl	t
var5:	kcl	k	aa	**n**	tcl	t	ey	kcl	k	tcl	t

Figure 13 Relaxation of misrecognized form toward canonical Shown are five variant pronunciations. The first is the observed phoneme string, now added as a valid pronunciation of the misrecognized word. Each subsequent variant represents one incremental step transforming the observed phoneme string back toward the canonical form (in no particular order). All five variants correct the misrecognition; var5 shows the smallest distance from the canonical form, and relaxation stops (since the next change would reproduce the canonical form which is known not to lead to correct recognition in this case). var5 leads to the suggested protorule (see § 4.6) that /ae/ may be substituted by /ey/ in the context shown (which might be abstracted to a context of, e.g., 'between two stop consonants').

string is then possible. Thus, e.g., in the sentence "UAL 7 14 maintain heading 0 4 0", misrecognized as "USA 1 14 maintain heading 0 4 0", the raw (bigram) phonemic transcription:

sil v tcl t sil v y ah ey s eh v ah zh f ao r dcl jh iy ...

can be segmented as:

sil v tcl t sil v y ah ey	s eh v ah zh	f ao r dcl jh iy ...
united	7	14

In practice, for this technique (and most others suggested here) it is the case that the recognizer is told which word is misrecognized. This provides a bracketed segment of the phoneme sequence which must correspond to the misrecognized word.

Step 2 requires identifying, within the isolated segment, the likely boundaries for the word. In the example above, the word 'united' (canonically: /y uw n ay tcl t {ah | ih} dcl d/) must be located within /sil v tcl t sil v y ah ey/. The subsegment /y ah ey/ is identified as the best fit, and the alignment shown in Figure 14 is established.

Step 3 can be performed as a straightforward left-to-right correction of errors, with each successful correction retained and each failure reverted to its original state. The left-to-right sweep can be performed as often as is necessary to arrive at the 'best' variant, determined as that variant differing least from the canonical form, correcting the misrecognition, and being a plausible pronunciation. This last condition arises in cases where two (or more) successful variants are derived satisfying the first two criteria as, e.g., /dcl d eh l t ah/ and /dcl d eh l tcl ah/ for 'DAL.' Fortunately, the belief score embodies these three criteria, and it suffices to select the variant with the best belief score:

	Pronunciation	Log Prob. Pron.	Log Prob. Acoust.	Log Belief
(i)	dcl d eh l DEL t ah	-1.216358e+01	-4.640420e+04	-4.641636e+04
(ii)	dcl d eh l tcl DEL ah	-1.216687e+01	-4.638151e+04	-4.639368e+04

Figure 14 shows this being performed for 'united.'

It may happen that even when the observed phoneme string is introduced at the beginning of this step as a variant pronunciation that recognition is not corrected. This indicates that the segment flagged as containing the misrecognized word is incorrectly positioned with respect to the signal, or, that the error is caused by an effect of the previous and/or following word. This is almost always a coarticulation effect, commonly observed in cases like "JAL 7..." where the word final /z/ of 'JAL' merges with the word initial /s/ of '7'. In such a case the procedure first tries to get a good variant pronunciation for the misrecognized word W_i before beginning on W_{i+1} or W_{i-1}, as the case may be. A final pass of relaxation might then be undertaken on the combined word pair, using the best obtained variant for each, in an attempt to determine if any better variant pair is obtainable.

It certainly seems apparent that introducing variant pronunciations which are 'close' (e.g., in edit distance) to canonical forms, as opposed to the more distant

canonical:	y	uw	n	ay	tcl	t	ah/ih	dcl	d	-4.327788e+04	
observed:	y			ah			ey			-4.309143e+04	
var1:	y	**uw**		ah			ey			-4.312302e+04	+
var2:	y	uw	**n**	ah			ey			-4.317906e+04	+
var3:	y	uw	n	**ay**			ey			-4.317000e+04	+
var4:	y	uw	n	ay	**tcl**		ey			-4.320013e+04	+
var5:	y	uw	n	ay	tcl	**t**	ey			-4.327788e+04	×
var6:	y	uw	n	ay	tcl		**ah**			-4.327788e+04	×
var7:	y	uw	n	ay	tcl		**ih**			-4.321269e+04	+
var8:	y	uw	n	ay	tcl		ih	**dcl**		-4.320510e+04	+
var9:	y	uw	n	ay	tcl		ih	dcl	**d**	-4.322909e+04	+

Figure 14 Iterative left-to-right correction of misrecognized form toward canonical Shown are the nine steps of a first left-to-right pass adjusting the observed phoneme string, which differs from the canonical by six deletions and two substitutions. No subsequent passes were needed in this case as the result of the first pass is one deletion away from the canonical form. The acoustic scores reported by the recognizer for each variant (log probability) include the score for the misrecognized string using the canonical pronunciations. Except for variants 5 and 6, each variant corrects the misrecognition, with variant 9 being best (smallest distance from canonical, 1 deletion). It is expected, as each variant is tried, that the acoustic score worsens with respect to the score for the observed phoneme string. Unusual in this particular example is the improvement in score observed between variants 2 and 3. The protorule (see § 4.6) inferred from this operation is that a /t/ may be deleted following a /tcl/: tcl t → tcl ε.

misrecognized strings, is preferable. This method of generating VPCs yields the one best variant given, as starting point, the observed phoneme string for the word. Automating the procedure described here was not explored in this work, although it might prove useful as a strategy for producing variant pronunciation candidates. One of the particular attractions of this technique, even performed as a manual exercise, is that it can handle cases where a misrecognition cannot be corrected with a single phoneme change, a limitation of other methods introduced.

4.4 The Phoneme Lattice

During recognition, as state sequence paths are built and extended, it is possible to snap-shot the state of the above-threshold paths in a useful way. That is, at each time frame one can note which phonemes are active in some path. A table of such activations with respect to frame number (time), called a *lattice*, may be constructed; see Figure 15. In the figure, where the word 'three' occurs, one sees that in addition to the canonical pronunciation /th r iy/, many other phonemes are active through part or all of the interval (t_b, t_e) for word 'three.' Thus, for example, a variant pronunciation /tcl t r iy/ such as an Irish speaker of English might say, is feasible since all of these phonemes are also active in (t_b, t_e).

This raises two possible uses for the phoneme lattice. In either case, the lattice is used in conjunction with other information, in order to provide a set of variant pronunciation candidates. If one already has some means of generating variant pronunciation candidates, one can use the lattice as a filter to cull from a generated set those candidates for which there is no acoustic evidence. Alternatively, one may use the lattice to hypothesize a set of variant pronunciations, based on acoustic evidence, and then apply some set of constraints to eliminate those sequences of phonemes which are implausible.

Use of the phoneme lattice as a post-variant generation filter is discussed in other sections of this chapter. Use of the lattice as a primary means of generation is not examined here.

4.5 A Substitution Rule Approach

A simple approach to generating variant pronunciation candidates that does not use observation data directly, as the abovementioned techniques did, is to begin with pronunciations differing from the canonical form by one phoneme. A common observation when working with the relaxation-to-canonical technique described above (§ 4.3) was that substitution of a single phoneme was usually† successful at correcting misrecognition. Moreover, the canonical phoneme and the phoneme with which it is substituted are nearly always members of the same 'class', i.e., a fricative is replaced by another fricative, a vowel by a vowel, etc. (see Table 9). This also suggests the basis for a simple generative technique for rating variant pronunciation candidates.

A simple class division of phonemes is shown in Table 4. One can introduce a special phoneme to represent deletion of a phoneme, making it a member of each class, thus making deletion a special case of substitution. One may also introduce special classes to allow modeling of particular classes of substitution, e.g., a class containing {tcl_t th} would permit this method to propose /tcl t r iy/ as a variant pronunciation for the word "three."

Admittedly, this approach only handles single phoneme errors. Yet, on an evaluation subset of ATS2 (26 sentences × 3 speakers), single phoneme substitutions and deletions account for 83.3% of the misrecognitions. A simple and inexpensive approach such as this one, able to correct over 80% of misrecognitions, is worthy of

† ≥ 70% of the time in various tests conducted on subsets of ATS2 and other corpora during the thesis work.

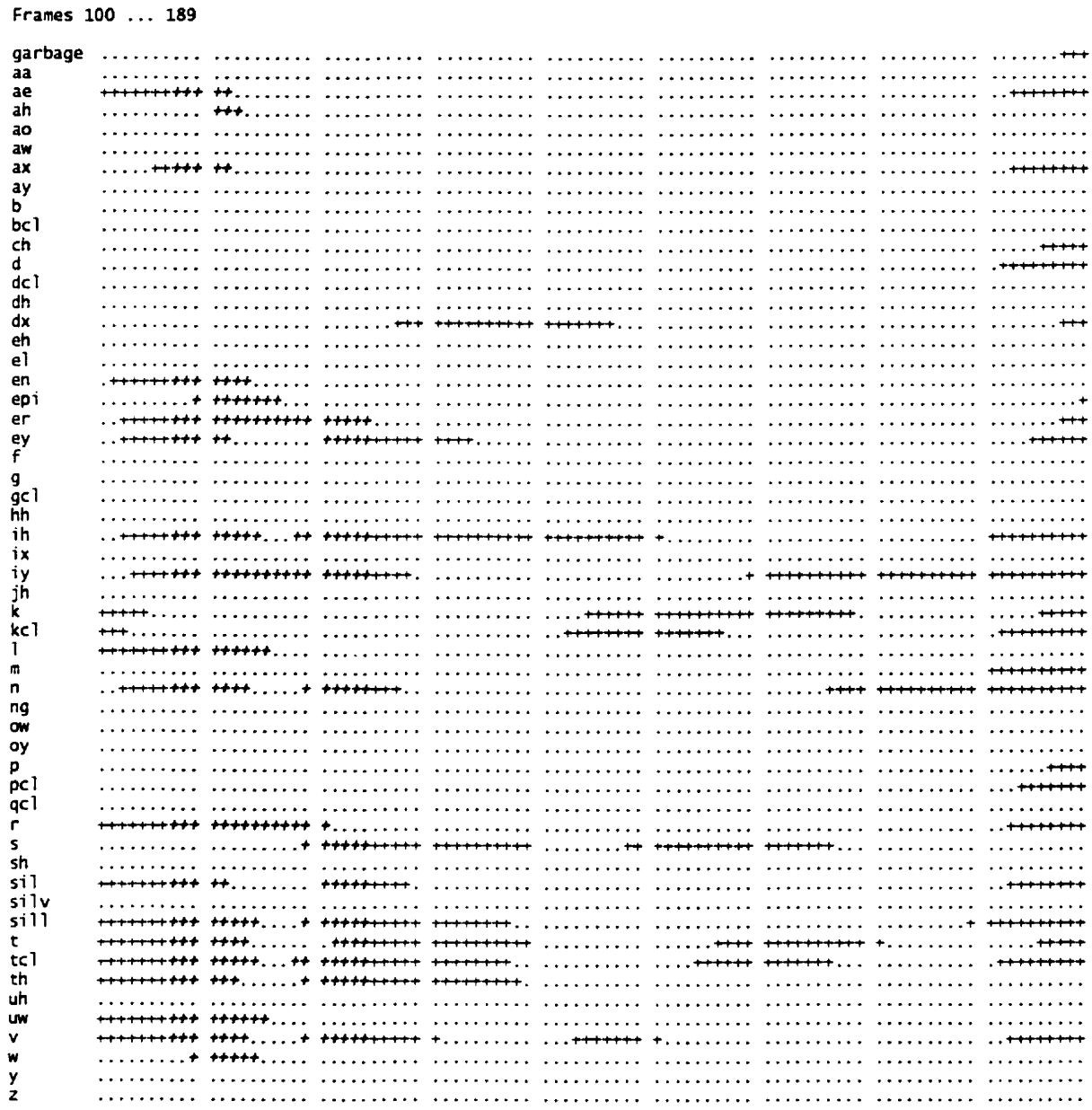

Figure 15 Sample Phoneme Lattice showing activation of different phonemes in frames ('+' → active, '.' → inactive). Shown is part of the lattice (frames 100 to 189); gaps appear between blocks of 10 frames. The highlighted region corresponds to the word 'three' in the sentence "Continental **3** 16 affirmative flight level 4 2 0." The 'garbage' model is an acoustic model trained so as to score highly on non-speech acoustic information, e.g., to match coughs, ticks, lip-smacks, etc., thereby, in effect, preventing a phoneme model from scoring highly enough to introduce misrecognition.

Table 4 Initial Phoneme-by-Class assignments

vowel	iy ih ey eh ae aa ax uw uh ow ao ah er ix
diphthong	ay oy aw
glide	y w
liquid	l r
nasal	m n ng
fricative	f v th dh s z sh zh hh
stopclosure	pcl bcl tcl dcl kcl gcl
stop	p b t d k g
affricate	ch jh

pursuit.

A simple algorithm for generating variants is then:

```
for each phoneme i in canonical pronunciation w_k

    for each phoneme j in the class of phoneme i

        generate one variant pronunciation in which
        phoneme i is substituted with phoneme j
```

Using such a scheme, the word 'contact' yields 72 variant forms, as shown in Table 5.

Even though the number of variants generated is linearly proportional to the length of the word (see Figure 16), this rule is sufficiently prolific at producing variants that some form of *filtering* may be desired to reduce the number of candidates. One available filtering mechanism is the lattice of active phonemes (see Figure 15) generated during recognition: it indicates which phonemes had above threshold activations (i.e., scored high enough to be kept as part of a path through the network of HMMs) for each time frame. Variant pronunciation candidates containing phonemes for which there is no acoustic evidence, i.e., which were never active in the interval (t_b, t_e), or even $(t_b-\varepsilon, t_e+\varepsilon)$, are culled from the list of candidates; see Table 6.

Table 5 Variant Pronunciation Candidates Using Within-Class Substitution The list below shows all variants generated using this simple technique for the word 'contact.' Note that the first listed variant is the canonical form for the word.

kcl k aa n tcl t ae kcl k tcl t	kcl k ah n tcl t ae kcl k tcl t	kcl k aa n tcl t ao kcl k tcl t
pcl k aa n tcl t ae kcl k tcl t	kcl k er n tcl t ae kcl k tcl t	kcl k aa n tcl t ah kcl k tcl t
bcl k aa n tcl t ae kcl k tcl t	kcl k ix n tcl t ae kcl k tcl t	kcl k aa n tcl t er kcl k tcl t
tcl k aa n tcl t ae kcl k tcl t	kcl k aa m tcl t ae kcl k tcl t	kcl k aa n tcl t ix kcl k tcl t
dcl k aa n tcl t ae kcl k tcl t	kcl k aa DEL tcl t ae kcl k tcl t	kcl k aa n tcl t ae pcl k tcl t
DEL k aa n tcl t ae kcl k tcl t	kcl k aa ng tcl t ae kcl k tcl t	kcl k aa n tcl t ae bcl k tcl t
gcl k aa n tcl t ae kcl k tcl t	kcl k aa n pcl t ae kcl k tcl t	kcl k aa n tcl t ae tcl k tcl t
kcl p aa n tcl t ae kcl k tcl t	kcl k aa n bcl t ae kcl k tcl t	kcl k aa n tcl t ae dcl k tcl t
kcl b aa n tcl t ae kcl k tcl t	kcl k aa n DEL t ae kcl k tcl t	kcl k aa n tcl t ae DEL k tcl t
kcl t aa n tcl t ae kcl k tcl t	kcl k aa n dcl t ae kcl k tcl t	kcl k aa n tcl t ae gcl k tcl t
kcl d aa n tcl t ae kcl k tcl t	kcl k aa n kcl t ae kcl k tcl t	kcl k aa n tcl t ae kcl p tcl t
kcl DEL aa n tcl t ae kcl k tcl t	kcl k aa n gcl t ae kcl k tcl t	kcl k aa n tcl t ae kcl b tcl t
kcl g aa n tcl t ae kcl k tcl t	kcl k aa n tcl th ae kcl k tcl t	kcl k aa n tcl t ae kcl t tcl t
kcl k iy n tcl t ae kcl k tcl t	kcl k aa n tcl DEL ae kcl k tcl t	kcl k aa n tcl t ae kcl d tcl t
kcl k ih n tcl t ae kcl k tcl t	kcl k aa n tcl t iy kcl k tcl t	kcl k aa n tcl t ae kcl DEL tcl t
kcl k ey n tcl t ae kcl k tcl t	kcl k aa n tcl t ih kcl k tcl t	kcl k aa n tcl t ae kcl g tcl t
kcl k eh n tcl t ae kcl k tcl t	kcl k aa n tcl t ey kcl k tcl t	kcl k aa n tcl t ae kcl k pcl t
kcl k ae n tcl t ae kcl k tcl t	kcl k aa n tcl t eh kcl k tcl t	kcl k aa n tcl t ae kcl k bcl t
kcl k DEL n tcl t ae kcl k tcl t	kcl k aa n tcl t DEL kcl k tcl t	kcl k aa n tcl t ae kcl k DEL t
kcl k ax n tcl t ae kcl k tcl t	kcl k aa n tcl t aa kcl k tcl t	kcl k aa n tcl t ae kcl k dcl t
kcl k uw n tcl t ae kcl k tcl t	kcl k aa n tcl t ax kcl k tcl t	kcl k aa n tcl t ae kcl k kcl t
kcl k uh n tcl t ae kcl k tcl t	kcl k aa n tcl t uw kcl k tcl t	kcl k aa n tcl t ae kcl k gcl t
kcl k ow n tcl t ae kcl k tcl t	kcl k aa n tcl t uh kcl k tcl t	kcl k aa n tcl t ae kcl k tcl th
kcl k ao n tcl t ae kcl k tcl t	kcl k aa n tcl t ow kcl k tcl t	kcl k aa n tcl t ae kcl k tcl DEL

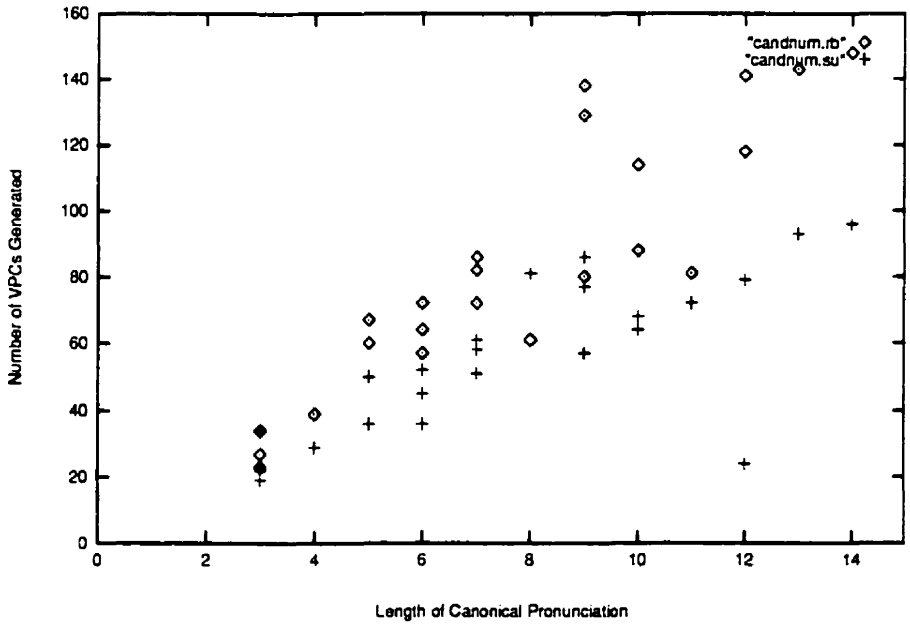

Figure 16 Relation of Number of Generated Variants to Length of Canonical Form The ♦ represent variants generated using the rule based method (see § 4.6); the + represent variants generated using the substitution rule method.

Table 6 Lattice Filtering of Substitution-Rule Generated Variants In recognition work reported elsewhere [58] variant pronunciations were developed on a set of 26 sentences in the ATS2 task uttered by three speakers. The table below shows, for words that were misrecognized, the number of variant pronunciation candidates generated using the 'substitute one phoneme with another of the same class' rule, and the number that remained after filtering with the phoneme lattice for the interval (t_b, t_e). In cases where the lattice eliminated all candidates, the technique exhaustively tests all of the originally proposed candidates, and often finds variant pronunciations successful in correcting the misrecognition. In those cases where 0 successful candidates were found, correction of the misrecognition was achieved by using (manually) the iterative transformation process described in § 4.3. Note that many of the misrecognized words are single digits or numbers; these remain a particularly challenging component of a vocabulary to recognize.

Word	Speaker ID	Number of Variants Generated	Number of Post-Lattice Variants	Number of Successful Variants
2	JA	23	9	1
2	VP	23	14	1
3	JA	19	11	1
3	VP	19	11	1
3	DS	19	3	2
5	JA	22	0	3
6	JA	45	12	2
7	JA	50	0	0
8	VP	23	7	0
9	DS	10	0	2
12	VP	36	8	5
12	DS	36	7	3
13	VP	42	27	1
15	VP	58	0	7
17	VP	75	0	5
19	VP	35	1	1
affirmative	VP	86	4	4
avianca	JA	81	11	3
avianca	VP	81	12	1
continental	JA	96	11	0
continental	DS	96	11	0
flight level	JA	64	0	17
roger	VP	33	19	3
speed bird	JA	86	0	3
traffic	VP	60	3	1
turn	VP	26	13	3
united	JA	114	0	2

4.6 Rule Based Variant Generation

A generalization of the substitution rule based method of § 4.5 is the use of a set of rules to guide construction of VPCs. One begins with the canonical pronunciation, as

before, but now searches a set of rules to find those that are applicable to the case at hand, and through application of the subset of found rules, generates distorted forms of the canonical pronunciation as VPCs.

The rules suggested by phoneticians and linguists should, in the general case, prove to be effective as a basis for creating variant pronunciation candidates. But, for particular recognition problems, as was pointed out earlier, the pronunciation recommended by a phonetician may not be the most frequent or most likely pronunciation encountered. The work of Tajchman, Jurafsky and Fosler [61] establishes probabilities for optional rules based on a particular corpus and so can potentially provide rule probabilities adapted to a particular task. That they independently computed rule probabilities on a different corpus (TIMIT) and found the probabilities were highly similar is a good reflection of the general applicability of the rules. Their technique relies on being provided phonological rules *a priori*, and subsequently learns how applicable the rules are as the respective rules' probabilities are determined.

An alternative approach is to begin with no rules and acquire rules through, e.g., training on a corpus of labeled recordings. An intermediate approach might begin using the simple substitution rule (§ 4.5) or iterative transformation (§ 4.3) to generate variants. From those candidates found to be successful, one may infer rules and so accumulate a set of rules that, at some point, is rich enough to be used by itself to suggest what variants to generate. For example, a comparison of the canonical form with the best variant (e.g., minimal edit and class distance) produced using iterative transformation suggests the rule:

n tcl t *ae* kcl k tcl t → n tcl t *ey* kcl k tcl t.

Such a formulation is refered to as a *protorule*, and it is entered into a collection of similar protorules.

In the more general case, beginning with no rules, one must first acquire protorules before being able to generate variants. The set of protorules consists of rule statements of observed (eligible) distortions of words. Once the set of protorules has

been acquired, one begins using it in recognition work and can thus begin assessing the utility of the individual protorules. With each protorule is kept a set of counts (number of times applied, successes, failures, etc.) from which probabilistic scores can be computed, and on the basis of which the (now) rules can be assessed and ranked. Thus, after an initial training run on some set of corpora to acquire protorules, and subsequent use of the protorules in a recognition training run, one has a set of rules to use to guide the generation of VPCs.

A training run on a suitable corpus (WSJ0) was performed to generate a collection of protorules. Rules were distilled from a set of 315 utterances in the WSJ0 task training sentences (35 sentences × 9 speakers were used). The procedure obtains, for some utterance *X*, the best observed phoneme segmentation provided by the WSJ0 bigram-based recognizer (see Appendix 2). It also uses the canonical phoneme string and word segmentation provided by a forced Viterbi alignment recognition. A weighted dynamic alignment between observed and canonical phoneme sequences is used to generate the protorules for that utterance. The weightings reflect the fact that different misalignments can, acoustically, have different costs. For example, a within-class substitution of one vowel for another has lower substitution cost than a substitution of a fricative for the same vowel. Protorules are generated without regard as to whether a particular word may have been misrecognized, but rather, for any difference between the observed and canonical phoneme strings. Each generated protorule is then compared against the existing set of protorules: if it is novel, it is introduced into the collection; otherwise the count of its occurrences is incremented to reflect another independent observation of the protorule.

Each of these protorules has a target phoneme as well as left and right contexts consisting of some number of predecessor and successor phonemes (context width). Special tokens were introduced to indicate, in these contexts, when a word boundary was crossed, as well as start and end of an utterance. In forming the set of protorules, the context width was chosen to be two. This resulted in the generation of

7,158 distinct protorules for the 315 sentences examined. Table 7 shows some examples typical of these protorules.

One notes as well that among the protorules acquired there are some suggesting substitutions or deletions that the substitution rule method of § 4.5 also described (e.g., the four substitution rules about target /s/). There are other distortions described by protorules that the substitution method could not have produced, e.g., the /r/ substitution, and the /y/ insertion before /uw/, a commonly observed distortion in speech. Coarticulation is captured in this set of protorules, e.g,. the last example of /n/. While the substitution method could also handle such coarticulation (since DEL is a valid phoneme in every class), it could not have provided any context information to suggest that the proposed candidate should be used only in certain contexts.

The training procedure is susceptible to alignment errors in the forced recognition. If the misalignment is not a consistent problem in recognition, then any protorules proposed as a consequence of the misalignment will be poorly substantiated. If the misalignments are consistent, then the more strongly supported protorules which result will reflect a true artifact of the recognizer, potentially a useful source of misrecognition correcting information.

The number of protorules, k, may be unwieldy. It may, consequently, be reduced using a compression procedure which selectively merges together protorules having, e.g., common left or right contexts, common contexts and common classes of target phoneme, etc. Note, for example, in Table 7, those cases that have /s/ as their target. These, all substitutions of /s/ with /z/, show that a significant reduction in the number of rules may be achieved by compressing rules of a common target, type and value with similar left and right contexts. These might be reduced to s $\rightarrow$ z in context:

<liquid> /aa/ /S/ <mid/lax vowel> <fricative>

Table 7: Typical Protorules learned on wsj0 subset The 'target phoneme' is the phoneme to which the protorule applies; 'type' is one of Insertion, Deletion or Substitution; 'alt value' is, for type S, the phoneme which is substituted for the target, and for type I, the phoneme inserted before the target (this column has no meaning for type D); 'context' shows the context (2 phonemes right and left) of the target. The special symbol '<|>' indicate a word boundary, so cross-word contexts are preserved in this representation. The last column reports the number of times this particular rule (same target, same right/left context and same word) was encountered.

Target Phoneme	Type (I,D,S)	Alt Value	Context						Num. of Obs.
aa	S	ae	d	ay	AA	cl	k		10
aa	S	ao	vcl	d	AA	cl	k		1
ae	I	y	vcl	g	AE	n	<|>	cl	1
d	D	-	ah	vcl	D	iy	<|>	ah	
iy	S	ey	ao	r	IY	<|>	ah	v	1
n	S	m	vcl	d	<|>	N	ao	r	1
n	S	m	dh	iy	<|>	N	ey	ch	1
n	S	m	dh	iy	<|>	N	ey	cl	2
n	D	-	w	aa	N	<|>	n	ae	1
r	S	eh	f	aa	R	m	z		1
s	S	z	l	aa	S	ah	f		1
s	S	z	r	aa	S	eh	s		3
s	S	z	g	ae	S	<|>	vcl	d	1
s	S	z	k	ae	S	cl	t		1
uw	I	y	vcl	d	UW	s	ih		2

or

<stop> /ae/ /S/ <closure (voiced or unvoiced)> <stop>

Variant pronunciation candidates are generated according to:

```
for each phoneme m in canonical pronunciation w_i

    for each protorule in ruleset

        if protorule applicable to m in context
            generate variant
            if this variant is novel
                add new variant to list
            else
                add this protorule's id to list of rules
                substantiating this variant
```

One may choose, of course, how a protorule is 'applicable': e.g., merely has the same target phoneme, or, same target + same contexts. One may also establish a

criterion such that a particular variant pronunciation candidate must be suggested by σ rules before being accepted. This affects the quality and quantity of the variant pronunciation candidates generated. As one relaxes constraints on the matching of context in a protorule to the context in the canonical pronunciation, it is evident that more rules will match and, consequently, more variant pronunciation candidates will be generated.

4.7 Performance Models as Variant Generators

The rule based approaches described above constitute a 'prescriptive' approach to generating variant pronunciation candidates. A different approach, inspired by the acoustic model architecture used in the recognizer, is to *model* the pronunciation of a word, providing probabilistic scores for each different pronunciation.

The *performance model* (see Figure 17) is an output generating automaton of the Mealy machine type. Transition and emission probabilities are learned from observations drawn from any suitable corpus. The pronunciation of a word represented using performance models embodies observed substitution, deletion and insertion events from observations of real speech. It may not include possible variants arising from idiosyncrasies of the particular recognizer, but these may be added. Nor does it include context sensitivity, e.g., the model will report the probability of a particular substitution as the pooled probability of that substitution from all contexts in which it was encountered.

A first set of performance models was trained on TIMIT. Substitution, insertion, and deletion counts were obtained for each phoneme by aligning ideal and observed phoneme transcriptions of each utterance. This was done by for each district (dr1 – dr8), whence global counts were derived. For the word 'contact' (see example in § 4.5), the performance model variants scoring above threshold probability are shown in Table 8 (compare with Table 5).

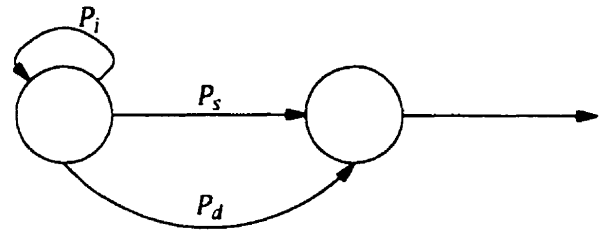

Figure 17 A phoneme performance model The transition and emission probabilities are learned from training material in a speech corpus. P_d is the probability that the phoneme being modeled is deleted (so only emits ε). P_i is the probability of an insertion before the phoneme being modeled and the associated emission probabilities, 1 per phoneme, are based on what was observed in the training data, i.e., where insertions occurred before the target phoneme. P_s is the probability that the phoneme being modeled is unchanged or substituted for some other phoneme, and the emission probabilities are based on observed substitutions for the target phoneme.

Selection of an acceptance threshold is essential as, without one, the full set of variant pronunciation candidates is unreasonably large (i.e., the full Cartesian product of observed substitutions). In the case of 'contact', 1,397,088,000 variants are possible in dr1 alone; with a threshold of $1e^{-5}$, 88 candidates are proposed.

In the absence of a clear suggestion as to a 'good' value for the threshold, a value such that the performance model generates roughly the same number of variants as more conventional methods is chosen. Examining the sets of variants produced by performance model and rule based methods shows relatively minor, but interesting, differences.

As described above (§ 4.1), the performance models trained on individual dialectic regions capture 'local' differences which affect variant pronunciation candidate scores; see Figure 18.

Table 8 Variant Pronunciation Candidates Using Performance Model Generator Substitution The list below shows all variants generated using a performance model trained on the dr1 district of TIMIT. Shown is the canonical form for the word, the substitution sets formed (i.e., non-zero substitutions for each of the canonical phonemes), and the variants so generated (88 in total, with a threshold set at $1e^{-5}$).

Word "contact" kcl k aa n tcl t ae kcl k tcl t

Lists of substitutions:

1 | 4| kcl = { t pcl kcl DEL }
2 | 7| k = { d dx p k q gcl DEL }
3 | 10| aa = { ih ix ax ah uh ao aa er axr DEL }
4 | 8| n = { l en eng m n ng nx DEL }
5 | 5| tcl = { q tcl kcl dcl DEL }
6 | 9| t = { d dx p t k q hv dcl DEL }
7 | 11| ae = { ih eh ae ix ax ah aa ey er ax-h DEL }
8 | 4| kcl = { t pcl kcl DEL }
9 | 7| k = { d dx p k q gcl DEL }
10 | 5| tcl = { q tcl kcl dcl DEL }
11 | 9| t = { d dx p t k q hv dcl DEL }

No threshold max number of variants: 1397088000

kcl k aa n tcl t eh kcl k tcl t
kcl k aa n tcl t ae kcl k tcl t
kcl d aa n tcl t ae kcl k tcl t
kcl p aa n tcl t eh kcl k tcl t
kcl p aa n tcl t ae kcl k tcl t
kcl k ih n tcl t ae kcl k tcl t
kcl k ah n tcl t eh kcl k tcl t
kcl k ah n tcl t ae kcl k tcl t
kcl k ah ng tcl t ae kcl k tcl t
kcl k uh n tcl t eh kcl k tcl t
kcl k uh n tcl t ae kcl k tcl t
kcl k uh ng tcl t eh kcl k tcl t
kcl k uh ng tcl t ae kcl k tcl t
kcl k ao n tcl d eh kcl k tcl t
kcl k ao n tcl d ae kcl k tcl t
kcl k ao n tcl t ih kcl k tcl t
kcl k ao n tcl t eh kcl k tcl d
kcl k ao n tcl t eh kcl k tcl t
kcl k ao n tcl t eh kcl k dcl t
kcl k ao n tcl t ae kcl k tcl d
kcl k ao n tcl t ae kcl k tcl t
kcl k ao n tcl t ae kcl k dcl t
kcl k ao n dcl t eh kcl k tcl t
kcl k ao n dcl t ae kcl k tcl t
kcl k ao ng tcl d ae kcl k tcl t
kcl k ao ng tcl t eh kcl k tcl t
kcl k ao ng tcl t ae kcl k tcl d
kcl k ao ng tcl t ae kcl k tcl t
kcl k aa l tcl t ae kcl k tcl t
kcl k aa m tcl t eh kcl k tcl t
kcl k aa m tcl t ae kcl k tcl t
kcl k aa n tcl d eh kcl k tcl t
kcl k aa n tcl d ae kcl k tcl d
kcl k aa n tcl d ae kcl k tcl t
kcl k aa n tcl p eh kcl k tcl t
kcl k aa n tcl p ae kcl k tcl t
kcl k aa n tcl t ih kcl k tcl t
kcl k aa n tcl t eh pcl k tcl t
kcl k aa n tcl t eh kcl p tcl t
kcl k aa n tcl t eh kcl k tcl d
kcl k aa n tcl t eh kcl k tcl p
kcl k aa n tcl t eh kcl k tcl t
kcl k aa n tcl t eh kcl k tcl k
kcl k aa n tcl t eh kcl k tcl dcl
kcl k aa n tcl t eh kcl k dcl t
kcl k aa n tcl t eh kcl gcl tcl t
kcl k aa n tcl t ae pcl k tcl t
kcl k aa n tcl t ae kcl d tcl t
kcl k aa n tcl t ae kcl p tcl t
kcl k aa n tcl t ae kcl k tcl d
kcl k aa n tcl t ae kcl k tcl p
kcl k aa n tcl t ae kcl k tcl t
kcl k aa n tcl t ae kcl k tcl k
kcl k aa n tcl t ae kcl k tcl dcl
kcl k aa n tcl t ae kcl k kcl t
kcl k aa n tcl t ae kcl k dcl t
kcl k aa n tcl t ae kcl gcl tcl t
kcl k aa n tcl t ah kcl k tcl t
kcl k aa n tcl t aa kcl k tcl t
kcl k aa n tcl t ey kcl k tcl t
kcl k aa n tcl t er kcl k tcl t
kcl k aa n tcl k eh kcl k tcl t
kcl k aa n tcl k ae kcl k tcl t
kcl k aa n tcl dcl eh kcl k tcl t
kcl k aa n tcl dcl ae kcl k tcl t
kcl k aa n kcl t ae kcl k tcl t
kcl k aa n dcl t eh kcl k tcl t
kcl k aa n dcl t ae kcl k tcl t
kcl k aa ng tcl d eh kcl k tcl t
kcl k aa ng tcl d ae kcl k tcl t
kcl k aa ng tcl p ae kcl k tcl t
kcl k aa ng tcl t ih kcl k tcl t
kcl k aa ng tcl t eh kcl k tcl d
kcl k aa ng tcl t eh kcl k tcl t
kcl k aa ng tcl t eh kcl k dcl t
kcl k aa ng tcl t ae kcl k tcl d
kcl k aa ng tcl t ae kcl k tcl p
kcl k aa ng tcl t ae kcl k tcl t
kcl k aa ng tcl t ae kcl k dcl t
kcl k aa ng tcl t ah kcl k tcl t
kcl k aa ng tcl t aa kcl k tcl t
kcl k aa ng tcl t ey kcl k tcl t
kcl k aa ng dcl t eh kcl k tcl t
kcl k aa ng dcl t ae kcl k tcl t
kcl k er n tcl t eh kcl k tcl t
kcl k er n tcl t ae kcl k tcl t
kcl gcl aa n tcl t eh kcl k tcl t
kcl gcl aa n tcl t ae kcl k tcl t

Table 9 Performance Model Substitution and Insertion Statistics Drawn from all eight 'dialectic districts' over which the TIMIT corpus is collected, the table shows how often a given phoneme is observed to occur as itself as opposed to how often it appears as a member of its own or some other phonemic class. The classes are fairly broad: Vowel, Liquid/Glide, Nasal, Fricative, Stop (either closure or consonant). The insertions column shows the distribution of occurrences of insertions by class, e.g., for /aa/, 189 of its 3,653 occurrences had a stop inserted before the /aa/.

		Occurs	As	As Nonself					Insertions				
Phoneme	Class	(Total)	Itself	V	LG	N	F	S	V	LG	N	F	S
aa	V	3653	2629	999	1	0	0	0	4	46	2	2	189
ae	V	6141	3932	2116	1	0	0	0	18	113	11	5	1013
ah	V	11582	2014	7840	86	0	0	0	77	118	23	20	461
ao	V	3342	2291	1032	0	0	0	0	7	33	2	209	443
aw	V	805	716	73	2	0	0	0	2	21	0	1	82
ay	V	2519	2352	155	0	0	0	0	2	22	2	0	167
b	S	2474	2151	0	0	0	0	7	3	60	0	1	0
bcl	S	2474	1878	0	0	0	0	10	7	9	6	8	4
ch	F	813	757	0	0	0	41	0	0	1	0	0	640
d	S	6402	3404	0	0	0	0	834	29	31	3	1	2
dcl	S	6402	3993	0	0	0	0	55	24	20	4	0	14
dh	F	3211	2807	0	0	0	178	0	6	82	4	3	4
eh	V	3054	2604	430	0	0	0	0	11	23	2	3	241
er	V	3740	1432	2077	168	0	0	0	10	25	2	13	86
ey	V	2156	2059	93	0	0	0	0	1	23	0	4	81
f	F	2192	2167	0	0	0	10	0	12	52	2	6	9
g	S	2373	1985	0	0	0	0	20	2	11	0	0	0
gcl	S	2373	2177	0	0	0	0	8	4	0	0	2	1
hh	LG	2230	869	3	20	1	0	0	13	46	2	4	6
ih	V	7744	3677	3761	18	0	0	0	31	141	15	39	538
iy	V	7177	6126	974	10	0	0	0	13	35	7	10	158
jh	F	1040	957	0	0	0	57	0	0	3	8	0	795
k	S	6376	4811	0	0	0	0	48	1	29	1	0	0
kcl	S	6376	5795	0	0	0	0	28	38	7	2	7	9
l	LG	6847	5725	5	924	0	0	0	41	69	11	19	7
m	N	4014	3814	4	9	122	0	0	14	393	3	9	26
n	N	8977	6867	8	6	1891	0	0	43	156	3	0	31
ng	N	1258	1129	1	1	118	0	0	1	2	0	2	1
ow	V	2160	2036	108	6	0	0	0	4	13	0	3	139
oy	V	763	666	96	0	0	0	0	30	3	0	0	250
p	S	2946	2551	0	0	0	0	27	0	14	0	0	0
pcl	S	2946	2596	0	0	0	0	18	6	0	2	3	1
r	LG	8449	6141	653	35	20	0	0	28	61	12	11	13
s	F	7556	7101	0	0	0	217	0	14	86	9	8	194
sh	F	2141	2102	0	0	0	17	0	2	59	0	0	28
t	S	10263	4240	0	0	0	0	2911	35	39	4	3	1
tcl	S	10263	5748	0	0	0	0	155	34	8	14	7	7
th	F	616	584	0	0	0	11	0	0	39	1	0	10
uh	V	564	336	208	0	0	0	0	5	4	0	1	6
uw	V	3363	514	2795	2	1	0	0	21	20	7	41	43
v	F	2064	1937	0	0	0	19	0	2	14	1	0	0
w	LG	3192	3086	1	16	2	0	0	19	708	2	2	32
y	LG	2066	1644	5	2	1	0	0	3	27	0	11	18
z	F	4142	3572	0	0	0	357	0	3	63	0	1	27
zh	F	82	68	0	0	0	13	0	0	1	0	0	9

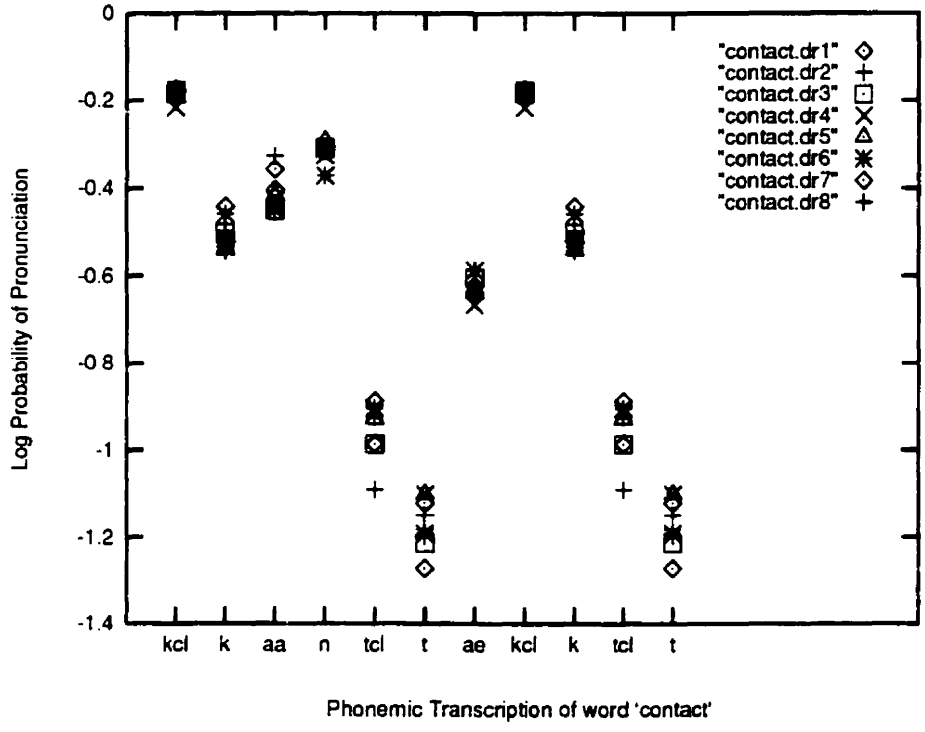

Figure 18 Dialectic Variation in Candidate Probabilities For the word 'contact', the figure shows the probability of observing each of the phonemes in the canonical pronunciation in each of the eight districts in which data were collected. Note that some phonemes are subject to greater variation than others. See also Figure 11.

4.8 Summary

Based on all of the above, it is possible to formulate a general procedure for obtaining rules to generate variant pronunciations:

1. *a priori* knowledge is used to formulate plausible phoneme substitutions, insertions, and deletions in terms of a set of possible replacements for each phoneme.
2. these sets are refined by the results of using performance models.

3. new rules involving left and right contexts are inferred from examples of effective variant pronunciations able to correct recognition errors and, perhaps, consistent with human expert expectations.
4. rules are generalized by factoring or introducing symbols for phoneme classes into the left and right contexts.

5. Recognition Using Variant Pronunciations

The previous chapter presented a few simple methods for generating variant pronunciations of a word. This chapter is concerned with how these methods can be put to use in improving recognition accuracy. Specifically, it looks at how the PD can be augmented in an automatic fashion with a reasonable set of variant pronunciations, the ultimate aim of which is to improve recognition accuracy. The chapter begins with a look at the data structure central to this work, then describes the procedure used to acquire variants, reports results of recognition tests using the pronunciations so acquired, and concludes with a look at what the successful variant pronunciations indicate about making variants.

5.1 The Pronunciation Dictionary

The pronunciation dictionary begins by containing the canonical pronunciations for each word in the task. These pronunciations were taken from the CMU 100,000 word pronunciation dictionary [12], which provides multiple canonical pronunciations for only very few words†. The intention is that this PD will grow as it acquires new variant pronunciations for words.

The PD is maintained as an ordered list of entries of this form:

word label pron class success fail used

where:

word is the orthographic (or other convenient) form of the word,

† For example, the pronunciation of the digit 0 is provided as /z ih r ow/ or /z iy r ow/.

label is a structured string which is:

- null if this entry is the only canonical form for the word,
- a single digit if there are > 1 (and < 10) canonical pronunciations of the word. In the ATS2 task, no word has more than 2 canonical pronunciations,
- a number followed by an alphabetic string if this is a variant pronunciation, e.g., '3AG.' The number identifies which method generated the variant pronunciation:

  1 rule-based (see § 4.6)

  2 substitution rule based (see § 4.5)

  4 coarticulation model (see § 5.4)

  8 iterative transformation (see § 4.3)

  16 other

  Pronunciations that arise from more than one method have a numeric subfield which is the result of ORing the contributing methods' tags, e.g., method 3 denotes a pronunciation generated by both rule-based and substitution rule based methods. The alphabetic string identifies which particular variant pronunciation this is using the convention A, B, ... , Z, AA, AB, ... , AZ, BA, BB, ... BZ ... and so on.

pron is the pronunciation of the word (currently expressed in TIMIT units),

class identifies the 'pedigree' of the pronunciation, with 0 denoting a canonical pronunciation. Variant pronunciations that have been generated and are undergoing evaluation belong initially to class 3, but may end up in class 1, 2 or 4 (see below).

success a count of the number of times this pronunciation has been used that resulted in word being correctly recognized,

fail — a count of the number of times this pronunciation has been used that resulted in the word being recognized incorrectly, i.e., the word for which this is the pronunciation was recognized but a different word should have been recognized,

used — a count of the total number of times this pronunciation has been used (redundant at present since it must be the sum of the two previous fields).

A variant pronunciation may be demoted in class if it is deemed 'too unlikely' a pronunciation (independently of the acoustics). If, for pronunciation w_{i_j}, $P(w_{i_j} \mid W_i) < \theta$, the pronunciation is rejected, i.e., is demoted to class 4. The threshold, θ, is $\theta = \alpha \frac{1}{N} \sum_{k=1}^{N} P(w_{i_{kc}} \mid W_i)$ where $w_{i_{kc}}$ is the k^{th} of N canonical pronunciations of word i. At present α is chosen, arbitrarily, as 10^{-36}.

The probability of the particular pronunciation, $P(w_{i_j} \mid W_i)$ is determined using some set of statistics on pronunciations. At present the pronunciations provided with TIMIT serve as the basis of these statistic (see § 4.7). Used for the results cited in this chapter was a merged set of all eight of TIMIT's dialectic regions.

Of variants that remain in class 3 after the above filtering step, those that are found in subsequent recognition tests to be highly successful may be promoted to class 2. If a pronunciation is observed to be successful across many speakers it may be promoted to class 1. No variant will ever be made a class 0 PD entry.

For ATS2 the canonical PD contained 179 entries. These pronunciation entries, with a file expressing the task syntax, are combined by a program to produce the stochastic finite state automaton used by the recognizer (see § 2.3). A consequence of this organization is that changes to the PD have no effect upon recognition until the pronunciations and syntax are 'compiled' into a new network.

5.2 Training

The purpose of training is to acquire alternative pronunciations for words which have been misrecognized when only the canonical pronunciations appearing in the initial PD are available. Table 10 presents a summary of recognition scores on the training, evaluation and test sets of the ATS2 task using only canonical pronunciations; these serve as the base reference to which all other results presented here are to be compared.

Training was performed on a subset of the ATS2 task designated as a training set (3 speakers × 47 sentences), and later on the evaluation set (9 speakers × ≈ 26

Table 10 Summary of Canonical Recognition Rates on ATS2 Task The table presents error rates for Sentence and Word recognition on the three partitions of the ATS2 set. The dialect classifications are approximate, and are NAE - native American-English speaker, NAE-MW native American-English speaker with strong U. S. Mid-West accent, NAE-F native American-English fast speaker, QF native Québec French speaker, and MIX refers to a speaker whose mother tongue is not English, but who learned English in both the UK and the UAR.

Speaker ID	Dialect Class	Phoneme Rate	Train Set		Eval Set		Test Set	
			S	W	S	W	S	W
JA	NAE	11.9	10.87	2.78	12.00	2.16	9.09	0.97
MS	NAE	9.8	27.27	9.11	—	—	—	—
PV	NAE-MW	10.5	40.48	14.29	—	—	—	—
ZA	light Arabic accent	11.1	—	—	14.29	2.81	25.00	4.62
JB	NAE-F	12.9	—	—	38.46	6.25	45.45	10.68
MB	QF	11.7	—	—	20.00	2.70	11.11	1.23
DS	NAE-F	12.7	—	—	30.77	5.00	30.00	5.75
QM	NAE	10.0	—	—	20.00	6.44	11.11	2.78
VP	strong Bangladeshi accent	10.2	—	—	57.69	22.50	70.00	18.39
TS	MIX	11.8	—	—	24.00	3.06	20.00	4.60
VV	moderate Indian accent	10.9	—	—	57.14	13.90	44.44	17.11

sentences). A rough classification of the speakers by dialect appears in Table 10. The rate of speech was highly variable across the speakers; the table shows speaking rate measured in phonemes per second averaged over the evaluation set (except for speakers MS and PV averaged over training set).

Training is performed beginning with a pure canonical version of the PD. Successful variants are accumulated as a set of 'preferred pronunciations' which may be saved (separately) following training. Thus a training session may be run for an individual speaker, a group of speakers, or an entire population of speakers, with the variants favoured by the user group (of one or more) kept distinct.

The procedure followed for training is:

```
for each utterance in the training set
        perform recognition on this utterance
        if recognition was incorrect
                display word and phoneme segmentation
                prompt for correction to make
                generate variant pronunciation candidates for correction
                do
                        build new task SFSA with all eligible variants in parallel
                        re-run recognizer
                        if recognition is correct
                                increment count of 'winners'
                                mark this pronunciation ineligible
                until no more winners found or maximum count of winners reached
update list of preferred-pronunciations
```

The displaying of the word and phoneme segmentation when soliciting the correction from the user is an anachronism from previous versions of the training

procedure. In fact, as currently implemented, it suffices merely to indicate which word in the sentence is to be corrected. Figure 19 shows a sample of the training procedure being used.

Testing recognition accuracy is done using a dynamic alignment scoring algorithm which compares the recognizer's hypothesized word string with the transcription obtained from the signal file.† The algorithm provides the number of insertions, deletions and substitutions needed to transform the observed (recognizer output) string into the reference (transcript) string. The algorithm is based on string comparison alone, so provides an edit distance, although this may not always be truly representative of the errors that occurred. For example, in Figure 19, the word VRG is misrecognized as the pair of words 'reduce speed'; this will be counted as two errors, a substitution and an insertion, rather than a single substitution of one word for two other words. In any case, use of this scoring method to count errors is made in a consistent and egalitarian way across all recognition experiments so meaningful comparisons can be made.

Two distinct training series were performed. The first used only the three speakers in the originally designated training set, with one set of preferred-pronunciations kept for all speakers. These variants were tested on that training set, and on the evaluation set of nine speakers (eight of whom were different from training set speakers). The second run was performed on the nine speakers from the evaluation set individually, with preferred-pronunciations kept for each speaker. These variants were then utilized in recognition experiments on the test set of 9 speakers × ≈ 11 sentences (same nine speakers as evaluation set). The outcome of the second training exercise is summarized in Table 11, below.

† The signal files are stored using NIST's SPHERE format which allows headers to contain arbitrary fields. Each signal file in ATS2 contains a transcript of what the speaker in fact said, and the prompting text for the sentence. For more information on the corpus, see Appendix 1.

```
SCORE:84:*-7.024306e+04
COMPARING REFERENCE:   VRG 2 11 maintain heading 3 3 0 degrees
TO RECOGNIZED:         reduce speed 2 1 3 3 indicated 4 sequencing 2 runway 0
RAW:                   reduce@2 speed 2 1 3 3 indicated@2 4 sequencing 2 runway 0@2
11 | ids: (3 / 0 / 8) <84
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
ERROR IN RECOGNITION
84:     VRG 2 11 maintain heading 3 3 0 degrees

 1:     160 640 @sil            13:    57600 60960 3
 2:     640 6880 @sil           14:    60960 63520 @sil
 3:     6880 9280 @sil          15:    63520 80640 indicated@2
 4:     9280 16000 reduce@2     16:    80640 84480 4
 5:     16000 30880 @sil        17:    84480 88320 @sil
 6:     30880 33440 speed       18:    88320 98400 sequencing
 7:     33440 36960 2           19:    98400 100160 @sil
 8:     36960 38880 @sil        20:    100160 101920 2
 9:     38880 47200 1           21:    101920 102720 @sil
 10:    47200 52000 @sil        22:    102720 107520 runway
 11:    52000 56000 3           23:    107520 112320 0@2
 12:    56000 57600 @sil        24:    112320 116960 @sil

BIGRAM SEGMENTATION:
        160 640 sil     32320 34560 uw  57120 57280 tcl 81760 82720 ao  107520 111200 z
        640 4960 th     34560 36640 ow  57280 58240 t   82720 83680 r   111200 116960 sil
        4960 5600 v     36640 39040 qcl 58240 59360 ih  83680 87520 iy
        5600 7360 hh    39040 41760 l   59360 61600 iy  87520 90560 z
        7360 8800 uh    41760 43680 ae  61600 63360 hh  90560 91680 iy
        8800 10880 r    43680 44640 dh  63360 65120 uh  91680 94240 r
        10880 12800 ih  44640 45440 eh  65120 65600 dx  94240 94880 ah
        12800 13600 gcl 45440 46880 n   65600 67040 iy  94880 96960 l
        13600 14400 dh  46880 47040 dcl 67040 68800 uw  96960 97120 dcl
        14400 15040 ih  47040 47520 d   68800 73280 sil 97120 98080 d
        15040 16480 dx  47520 49120 th  73280 74720 th  98080 100320 sil
        16480 17120 ix  49120 51680 f   74720 75360 ih  100320 101760 iy
        17120 20160 v   51680 53440 epi 75360 76640 r   101760 102400 gcl
        20160 23520 th  53440 54080 m   76640 78400 iy  102400 103040 g
        23520 30720 sil 54080 56160 ey  78400 80960 th  103040 104640 r
        30720 32320 jh  56160 57120 n   80960 81760 y   104640 107520 iy

Enter corrections as b,exw or hit return if is all correct
b,e are word numbers, x is one of i, d, or s, w is the word that should be there (for S)
4960,14400sVRG
VARIANT: "/home/speech2/charles/speech/bin/pa-risc/dovar" "VRG" 52 variants
VARIANT: "/home/speech2/charles/speech/bin/pa-risc/rulevar" "VRG" 64 variants
VRG has 1 canonical pron(s)
Average canonical pronprob is -1.9491, making cutoff -84.8422
VPC LIST RETURNED HAS 96 entries
ICW: VRG*2CN is  1
ICW: VRG*3AI is  2
ICW: VRG*2CJ is  3
ICW: VRG*2BI is  4
ICW: VRG*1CE is  5
ICW: VRG*3AG is  6
ICW: VRG*2CG is  7
ICW: VRG*2BT is  8
ICW: VRG*2AE is  9
ICW: VRG*2K  is 10

REWROTE /home/speech2/charles/ATS2//Pronsets/v7A/PD.txt: put 448 wordpron entries
```

Figure 19 Output From a Training Run Shown is a sample of the output from a typical training run. Words having more than one canonical pronunciation are displayed as *word@nn* where *nn* denotes which particular canonical pronunciation was used. The misrecognized word, [air-

line name] VRG, is supplied in response to prompt with a guess as to where the utterance the correction is to be made. In the current version of the training procedure it is sufficient to specify only the word. The VARIANT procedure is invoked with each method requested, here resulting in two sets of generated variants: the former is from use of the substitution-based method, the latter from use of the rule-based method. Each of the lists is sorted and contains unique entries, but there may be duplicates between the two lists. In this case, it turns out that the merged list contains only 96 distinct variants. The list of winning variants follows, in order of highest scoring to lowest scoring, and is stopped when 10 successful variants have been identified.

Table 11 Successful Variant Pronunciations by Speaker The table shows the number of variants found to be successful (as class 3 variants) at correcting misrecognition by speaker and by method of origin.

Speaker ID	Winning Variants Generated Using		
	Rule-Based	Substitution	Combined
JA	6	3	2
ZA	11	0	6
JB	5	4	2
MB	0	7	3
DS	2	6	1
QM	1	14	2
VP	13	20	20
TS	1	7	2
VV	14	30	33

5.3 First Set

The objective of the training was to obtain from the set of all variants generated a subset of reliable pronunciations to add to the canonical PD. The PD so augmented would then be tested on the nine speakers in the evaluation set, many of whom are not native American-English speakers.

Initial training of variant pronunciations was performed on the 3 speaker × 47 sentence training set. All three speakers in this set are native American-English speakers who differ in speaking rate and dialect (see Table 10). As such, their pronunciations were expected to be more or less 'standard' for native American English speakers.

Variants were generated using both the rule-based and substitution methods described in chapter 4. For the former, the 'supported-by-rules-observed-*n*-times' threshold was set to 2. Note that the training procedure used for developing variant

pronunciations was slightly different from that described above. In this (earlier) version, rather than test all the candidates in parallel, each was tested individually and designated an 'in-context-winner' if it succeeded in correcting the misrecognized work in the context in which it appeared in the sentence. Table 12 shows typical behaviour of the rule-based method with different threshold values.

At the conclusion of the training phase, the winning pronunciations were added to the PD and recognition performed on the training set to measure behaviour and to look for variant pronunciations that induced errors (i.e., more than they corrected errors). Following elimination of offending variant pronunciations, the updated PD was tested on the evaluation set; see Table 13.

Examination of misrecognitions suggested many were the result of a formerly inoffensive variant pronunciation now inducing errors. Speakers whose speech deviated far from that of native American-English speakers appeared particularly ill-served by the set of variants augmenting the canonical PD.

5.4 Second Set

Reflection on the poor showing reported above suggested that it was, perhaps, overly ambitious to expect a small number of pronunciations from a small, non-representative (with respect to the evaluation set) set of training speakers to provide good corrective power. It seemed apparent that the approach should be less ambitious, and that variant pronunciations should be subject to more scrutiny before being made eligible for more widespread access in the PD. The notion of class for entries in the PD was introduced as a consequence (see § 5.1).

Certain recognition errors which arose during these recognition experiments were difficult or impossible to overcome. Some of this difficulty arose from between-

Table 12 Comparison of Rule-Based and Substitution Variant Generating Methods Success Rates Shown is the proportion of in-context-winner pronunciations to the total number of variants proposed by a variant generating method. The ICW rates are averages since, while variants may have been generated for more than one speaker, different speakers may have had different degrees of success (numbers of ICWs). The rates are computed relative to the total number of variants generated, not to those variants retained as class 3 pronunciations (see text).

	Substitution		Rule Based					
			$\theta = 2$		$\theta = 3$		$\theta = 4$	
Word	Num. of variants generated	% which are ICW (total)	Num. of variants generated	% which are ICW (avg)	Num of variants generated	% which are ICW (avg)	Num of variants generated	% which are ICW (total)
3	19	0.0	34	5.9	17	5.9	12	8.3
5	22	4.6	23	4.3	12	0.0	6	0.0
6	45	0.0	46	0.0	27	0.0	21	0.0
7	50	2.0	67	3.0	34	5.9	26	7.7
8	23	4.4	23	8.7	15	6.7	9	11.1
12	36	5.6	57	3.5	31	3.2	16	0.0
15	58	8.1	72	13.9	36	16.7	27	14.8
AAL	77	7.8	138	5.1	80	7.5	60	10.0
AVA	81	4.0	61	3.3	38	5.3	24	8.3
COA	96	0.2	148	0.0	84	0.0	60	0.0
DAL	51	15.7	82	4.9	44	9.1	30	10.0
JAL	79	8.9	141	5.7	83	9.6	53	9.4
KAL	93	7.9	143	6.3	81	9.9	55	10.9
UAL	57	0.0	80	0.0	50	0.0	35	0.0
VRG	52	9.6	64	14.1	38	18.4	36	25.0
affirmative	86	5.2	129	4.7	44	4.5	53	1.9
approach	45	15.6	72	6.9	44	6.8	31	9.7
cleared_for	68	20.6	88	9.1	52	5.8	36	8.3
contact	72	1.4	81	0.0	54	0.0	32	0.0
flight_level	64	5.6	114	14.9	56	17.9	36	25.0
reaching	36	2.8	60	1.7	35	2.9	27	3.7
to	23	4.4	23	13.0	16	0.0	12	0.0
EastTexas	24	20.8	118	5.1	65	6.2	45	4.4
Flann	29	6.9	39	2.6	19	5.3	13	0.0
Modena	61	47.5	86	0.0	44	0.0	30	0.0

Table 13 Summary of Correction Rates on Training and Evaluation Set Using the procedure described in the text, the first recognition results using the variant generation procedures described in chapter 4 are presented. Results are shown as the percentage change with respect to the canonical recognition rate, hence positive values imply the error rate increased, negative values that it decreased.

Speaker ID	Train Set				Evaluation Set			
	Rule-Based		Substitution		Rule-Based		Substitution	
	SER	WER	SER	WER	SER	WER	SER	WER
JA	-41.18	-66.25	-58.82	-68.75	-40.00	-42.86	-20.00	-14.29
MS	-58.82	-14.47	-76.47	-53.95	—	—	—	—
PV	-40.00	-30.12	-30.00	-28.92	—	—	—	—
ZA	—	—	—	—	-11.11	-3.85	0.00	-15.38
JB	—	—	—	—	-47.06	-40.54	-29.41	-47.30
MB	—	—	—	—	+33.33	+15.38	+33.33	+69.23
DS	—	—	—	—	+11.11	0.00	+11.11	+29.41
QM	—	—	—	—	+11.11	-10.71	-22.22	-14.29
VP	—	—	—	—	+13.33	+31.65	0.00	+18.99
TS	—	—	—	—	+50.00	+87.50	+33.33	+25.00
VV	—	—	—	—	-5.88	-10.42	0.00	+20.83
Average	-46.67	-36.95	-55.10	-50.54	+1.65	+2.91	+0.68	+8.02

word coarticulation effects, something the variant generating methods were not particularly suited to fielding. A better solution was to allow the introduction of *coarticulation* pronunciations into the PD. The most egregious offender of this type was observed in the sentence beginning "Japan Airlines 6 17 ...", for which the canonical transcription is: /jh ah pcl p ae n eh r l ay n z s ih kcl k s s eh v ah n tcl t iy n /. The fricatives occurring at the word boundaries were merged in 75.0% of the cases resulting in misrecognition. Two coarticulation pronunciations were introduced as a result: a 6_17 word and a JAL_6_17 word. Of the 75.0% where misrecognition occurred, 66.7% use the former pronunciation when it is made available. When both pronunciations are available, 83.3% of the misrecognition cases use JAL_6_17.

Coarticulation pronunciations can be generated automatically from the task grammar following simple rules for cases known to be susceptible to coarticulatory distortion. On the other hand, such pronunciations may be introduced (as here) on an 'on-demand' basis. This, too, may be done automatically: a misrecognized word may first be subjected to a rule-based test to see if it qualifies as a case susceptible

Table 14 Summary of Recognition Performance on Evaluation Set The upper section reports word error rates, the lower section string error rates, on the 9 speaker evaluation set. Canonical error rates are reproduced here from Table 10 for convenience. "Method 3" is the combination of methods 1 and 2, i.e., the combination of rule-based and substitution based variant generating methods. Similarly, Method 7 is everything in Method 3 with the addition of coarticulation pronunciations for certain combinations of words (see § 5.4). Method F is method 7 to which are added variants produced by iterative transformation. The rightmost column shows the percentage change in error (with respect to canonical) of the best of the methods used.

Speaker Id	Total Words	Base Canonical	Rule Based	Substitution Rule Based	Method 3	Method 7	Method F	Best Correction
JA	231	2.16	1.30	1.30	0.43	0.00	—	-100.00
ZA	178	2.81	0.00	1.12	0.00	0.00	—	-100.00
JB	240	6.25	5.83	5.83	5.00	4.58	0.42	-93.28
MB	222	2.70	2.70	2.25	2.25	2.25	—	-16.67
DS	240	5.00	4.58	3.75	3.75	3.33	—	-33.40
QM	233	6.44	6.44	0.86	0.86	0.86	—	-86.65
VP	240	22.50	9.58	7.08	5.42	5.00	—	-77.78
TS	229	3.06	0.87	0.44	0.00	0.00	—	-100.00
VV	187	13.90	3.74	2.67	2.67	1.60	—	-88.49
Average	222	7.20	3.89	2.91	2.26	1.96	0.42	-77.63

Speaker Id	Total Sentences	Base Canonical	Rule Based	Substitution Rule Based	Method 3	Method 7	Method F	Best Correction
JA	25	12.00	8.00	8.00	4.00	0.00	—	-100.00
ZA	21	14.29	0.00	9.52	0.00	0.00	—	-100.00
JB	26	38.46	34.62	38.46	30.77	26.92	3.85	-89.99
MB	25	20.00	20.00	16.00	16.00	16.00	—	-20.00
DS	26	30.77	26.92	26.92	26.92	23.08	—	-24.99
QM	25	20.00	20.00	8.00	8.00	4.00	—	-80.00
VP	26	57.69	30.77	34.62	26.92	23.08	—	-59.99
TS	25	24.00	4.00	4.00	0.00	0.00	—	-100.00
VV	21	57.14	23.81	23.81	23.81	14.29	—	-74.99
Average	24	30.48	18.68	18.81	15.16	11.93	3.85	-72.22

to coarticulatory distortion. If so, an appropriate coarticulation pronunciation for that word pair can be generated and tested before recourse is made to variant pronunciations.

In an effort to see just how close to perfect recognition automated methods could go, a separate recognition run was made using all of the above described methods, with the addition of pronunciations generated using the iterative transformation method described in § 4.3. Even if successful at improving otherwise troublesome errors, this method should be kept as a method of last resort since it generates

pronunciations that are sometimes highly specific to a particular speaker and a particular utterance. With the introduction of ways to distinguish pronunciations on the basis of class and method, pronunciations arising from iterative transformation became less objectionable to use.

Table 14 shows the effect on recognition of using the methods described above on the evaluation set of sentences; Table 15 shows the corresponding results for the test set of sentences.

Table 15 Summary of Recognition Performance on Test Set The upper section reports word error rates, the lower section string error rates, on the 9 speaker test set. Canonical error rates are reproduced here from Table 10 for convenience. "Method 3" is the combination of methods 1 and 2, i.e., the combination of rule-based and substitution based variant generating methods. Similarly, method 7 is everything in Method 3 with the addition of coarticulation pronunciations for certain combinations of words (see § 5.4). Method F is method 7 to which are added variants produced by iterative transformation. The rightmost column shows the percentage change in error (with respect to canonical) of the best of the methods used.

Speaker Id	Total Words	Base Canonical	Rule Based	Substitution Rule Based	Method 3	Method 7	Method F	Best Correction
JA	103	0.97	0.97	0.97	0.97	0.97	—	0.00
ZA	65	4.62	4.62	4.62	4.62	4.62	—	0.00
JB	103	10.68	10.68	10.68	10.68	10.68	9.71	-9.08
MB	81	1.23	1.23	0.00	0.00	0.00	—	-100.00
DS	87	5.75	4.60	5.75	5.75	5.75	—	-20.00
QM	72	2.78	2.78	0.00	0.00	0.00	—	-100.00
VP	87	18.39	9.20	9.20	9.20	9.20	—	-49.97
TS	87	4.60	2.30	1.15	1.15	1.15	—	-75.00
VV	76	17.11	7.89	7.89	7.89	7.89	—	-53.89
Average	84	7.35	4.92	4.47	4.47	4.47	9.71	-45.33

Speaker Id	Total Sentences	Base Canonical	Rule Based	Substitution Rule Based	Method 3	Method 7	Method F	Best Correction
JA	11	9.09	9.09	9.09	9.09	9.09	—	0.00
ZA	8	25.00	25.00	25.00	25.00	25.00	—	0.00
JB	11	45.45	45.45	45.45	45.45	45.45	36.36	-20.00
MB	9	11.11	11.11	0.00	0.00	0.00	—	-100.00
DS	10	30.00	20.00	30.00	30.00	30.00	—	-33.33
QM	9	11.11	11.11	0.00	0.00	0.00	—	-100.00
VP	10	70.00	40.00	40.00	40.00	40.00	—	-42.86
TS	10	20.00	20.00	10.00	10.00	10.00	—	-50.00
VV	9	44.44	33.33	33.33	33.33	33.33	—	-25.00
Average	9	29.58	23.90	21.43	21.72	21.43	36.36	-41.24

5.5 Summary

Consideration of the training method developed in this chapter, and of the two sets of experiments whose results appear here, suggests the following as a procedure for training of variant pronunciations, beginning with a set of canonical pronunciations.

```
for each new speaker

    for each misrecognition

        determine if this misrecognition may have between-word
        coarticulation as a cause and, if so, introduce a
        'coarticulation pronunciation' to the (now empty) set of
        variant pronunciation candidates

        to the (possibly still empty) set of VPCs introduce variant
        pronunciations generated using (at least) rule-based and
        substitution-based methods

        re-run recognition of the sentence with all VPCs in parallel until:
              (i) N successful variants have been observed,
              (ii) no successful variants are observed
        In the latter case, abandon further effort.

        introduce the 1 ≤ n ≤ N best successful variants into the
        set of speaker-specific pronunciations for that particular user
```

After some number of training speakers has been encountered, and at intervals thereafter, it will be desirable to scan the sets of speaker-specific pronunciations in an attempt to observe trends useful for developing, e.g., accent specific clusters of variants.

The next chapter discusses the results presented in this chapter, including what may be built with learned variants, and suggests some directions for work extending that which is presented here.

6. Observations, Future Work and Conclusion

This thesis looks at the question of whether it is possible to provide, automatically, alternate pronunciations to a recognizer's PD and so improve recognition performance. To do this, it proposed a mechanism composed of the following steps:

(1) correct misrecognized words by proposing alternate pronunciations as candidates for membership in the PD,

(2) filter the candidates so proposed to eliminate unlikely pronunciations given some set of statistics about pronunciations,

(3) test candidates that survive the previous step in recognition to obtain acoustic scores for them, and,

(4) retain the *N* best alternate pronunciations in a 'managed' PD.

This chapter discusses issues surrounding each of these points, the results the recognition system was able to realize, and looks briefly at possible future work based on that described here.

6.1 The Pronunciation Dictionary and Alternate Pronunciations

6.1.1 Adding to the PD

The PD is a data structure central to the working of the speech recognizer. It contains one, or occasionally (in the case of multiple canonical pronunciations) two or three, pronunciations per word in the task. There is an interest in keeping this structure small, as the recognizer search space grows in proportion to the PD size. As more words are introduced, the chance of confusing added words with existing canonical pronunciations and/or each other increases. Thus, an increase in PD size not only increases space and time costs, but can increase misrecognition as well.

On the other hand, if the canonical pronunciations alone were sufficient, there would be no need to introduce alternate pronunciations. It is, however, a plain fact that not every user of a recognition system will pronounce all words 'canonically,' hence there is the need to augment the purely canonical set of pronunciations.

To tradeoff these two constraints, the notion was adopted to update pronunciations 'on-demand,' i.e., when a misrecognition occurs.

The introduction of multiple pronunciation entries into the PD is not the only way of providing alternate pronunciations to a task. A popular technique is the use of pronunciation *models* [64] where an HMM models the phoneme sequences which may be used to pronounce a word. The principle advantage to adding multiple individual entries over modeling pronunciations is that the former is a 'lightweight' method. Both methods require a training step, but models have many parameters to train, and the training is typically done as a batch job prior to system use. The technique of using individual pronunciations is a much simpler training procedure and the results can be used immediately in recognition; moreover the entire process can be performed while the system is being used. It is also possible to provide, dynamically, alternate pronunciations to accommodate new speakers, even those whose speech is notably different from that found in the previous training material. Model based approaches are more ponderous in response to speakers with different accents or dialects, and unless they have been trained on pronunciations featuring the different accents and dialects, may not perform at all well, as the model parameters may not adequately represent characteristics of the new pronunciations. Finally, individual pronunciations developed using the methods described here (or in a similar fashion) are highly portable, and may be 'dropped-in' to virtually any recognizer at no cost. Pronunciation models may only be added to recognizers using compatible means of representing pronunciations.

Individual pronunciations may be grouped in any functionally meaningful way desired. For example, if a number of speakers of a common dialect have developed

the same (or possibly only similar) alternate pronunciations, those common pronunciations can be collected into a set of pronunciations known to be 'good' for that particular dialect. If one has a method of identifying a speaker's dialect or accent (see, for example, [6,22]) then a set of pronunciations known to be suited to that dialect/accent group can be applied to augment the canonical PD for that speaker's use of the recognizer.

6.1.2 Generating Alternate Pronunciations

Two principle methods of generating variant pronunciations were explored. The main design criteria for these methods were that they should be usable with 'live' recognizer data, i.e., used as someone is using the recognizer, and that they produce a small set of plausible variant pronunciations for a given word. The rules directing the generative process constitute *a priori* knowledge used by the variant generator. They are inferred from a data-driven process, i.e., the suggestions of what distortions to make to the canonical form to generate a variant pronunciation must be based on observed instances of such distortions.

The first method developed was that which performed within-class substitutions, one at a time. The introduction of a DEL phoneme made it possible to handle single substitution and deletion errors with this method. The method is attractive not only for its simplicity but also in that it is speaker, dialect and accent neutral. In its current form, no one within-class substitution is considered more likely than any other, hence variants suggested by this method are equally suited (or unsuited) to any speaker.

The rule-based method was developed shortly after to provide pronunciations that could not be proposed by the earlier method, and would reflect actual speech. In particular, this method can deal with insertion errors, in so much as such errors have been observed to occur in the training data.

The rules developed for generating variants may be trained on any suitable corpus so as to provide a truer reflection of distortions in the speech to be recognized. In all experimental work reported here, the rules were derived from WSJ sentences spoken by native American-English speakers. Such rules might be expected not to perform so well for speakers whose accent is not represented in the training corpus. Indeed, speakers VP and VV show exactly this: the relative contribution to the set of successful variant pronunciations made by method 1 is much lower in their cases than it is in cases of native American-English speakers like JA and DS.

The behaviour of the rule-based method is controlled by a number of parameters. One may control the selection of which rules to use in generating variants by specifying how exact a context match is desired. In all the work reported here, exact target phoneme matches were required, but no constraint was applied to left or right context. The method, as currently implemented, allows constraints to be applied to target phoneme (same or any), left and right contexts individually (same or any), the word from which the rule is drawn (same or any), and what error type (same or any). One might expect this list to be refined to include, for the target, left and right contexts (each individually) whether there is a match based on class of phoneme, where classes might be user supplied or generic. The point to the plethora of choices is that the set of rules invoked can be made more or less specific in content. The entire rule set was sufficiently small (just over 7,000 rules) that experimental work conducted here used rather lax constraints.

Another parameter affecting the size of rule set used to generate variants is a threshold of how many times a rule has been observed in the training data. Table 12 shows the effect of choosing different values for this threshold, θ corresponding to the number of distinct occasions on which the rule is observed. Good results were obtained with $\theta = 2$, suggesting that rules observed to occur only once in the training data do not make effective contributions to the set of rules used to generate variants.

Several useful operations (none explored here) might be performed on the rule set to provide more powerful rules, and reduce the size of the set. In some cases, two rules may be generated at the same or adjacent locations in a word. For example, in the WSJ sentence "The female produces a litter of two to four young in November and December", a commonly observed pronunciation for the word 'produces' was /pcl p er dcl d uw s ih z/, as opposed to the canonical /pcl p r oh dcl d uw s ih s/. Looking at only the first five phonemes, the rule training procedure recognizes this as a case where two changes have been made, and so proposes two separate rules. This is unfortunate since the rule based variant generator, as used in this work, will never propose the observed pronunciation since it involves (at least) two phoneme changes. It would be easy to merge the two rules:

(1) /pcl p *r* oh dcl d/: /r/ → /er/

(2) /pcl p r *oh* dcl d/: /oh/ → ε

into one which better represents the actual distortion:

/pcl p *r oh* dcl d/: /r oh/ → /er/

Each of these two methods is limited in that it only proposes variant candidates that differ by one phoneme from the canonical form. This need not have been the case, i.e., all rules, for either method, found applicable to a word could have been applied to all phoneme positions, thereby generating a variant pronunciation set with richer coverage, but also of exorbitant size. The choice to limit the methods to a single application of a rule per proposed candidate limits the set size to a manageable number. It was found to be a reasonable choice since the great majority of errors are found to be 'off-by-one-phoneme' errors.

One other method was developed but not implemented, that of iterative transformation. This method is able to handle arbitrary errors in the observed pronunciation. The choice not to make this an automatic part of the procedure was based primarily on the expectation that variant pronunciations produced using this method would be 'brittle' since they would be unduly inspired by the acoustics of an

individual sentence and speaker. Such pronunciations were, in early testing, found to be poor candidates for generalization. Nevertheless, the power of the method to correct misrecognition remained alluring, and the later introduction of individual speaker specific sets of variant pronunciations largely mitigated brittleness these variants showed.

It became apparent during recognition experiments that, in some cases where between word coarticulation was responsible for misrecognitions, no method was able to provide reasonable alternate pronunciations. For example, in the case cited in Chapter 5, "Japan Airlines 6 17 ...", the only credible variant of the misrecognized '6' was /ih kcl k/. While this corrected misrecognition, it did not do so as satisfyingly as the introduction of a new 'coarticulation-word' to the grammar did (see Figure 20). The suggestion is that, where appropriate, it is preferable to introduce a coarticulation-word than to perform extensive mutilation of the 'word caught in the middle.' While the results reported in Chapter 5 introduced coarticulation words only as needed, it may be preferable to introduce them task-wide (where possible) in a pre-processing step to reduce cases where the variant generation mechanism must be resorted to. Cases susceptible to coarticulation problems may be identified with a simple set of rules (e.g., cases of word final /s, z/ followed by word initial /s, z/). One may also supplement such a set with coarticulation effects used by a particular accent or dialect of speaker(s).

6.2 Pronunciation Likelihoods

Many of the generated variant pronunciations are implausible with respect to some reference, and can be eliminated from further consideration. The reference used throughout the research work was derived from TIMIT's dialect-region pronunciations (see Appendix 1). The eight dialect regions over which TIMIT sentences were recorded were merged into one common set, on which statistics for insertions, substitutions

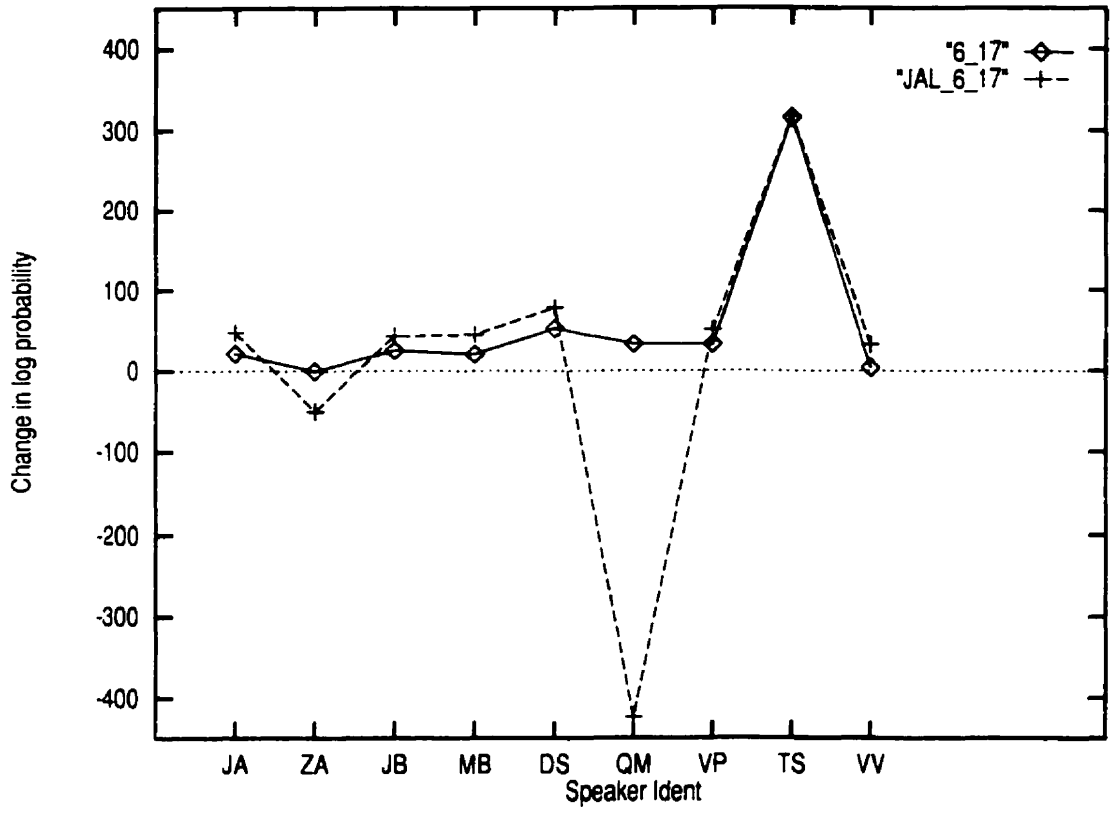

Figure 20 Effect on Acoustic Score of a Coarticulation Pronunciation The figure shows the change in acoustic score using 'coarticulation words' "6_17" and "JAL_6_17" with respect to the acoustic scores obtained using discrete words "JAL 6 17." For each speaker the value plotted in the difference in log probability obtained as (acoustic score using coarticulation word) - (acoustic score using discrete words), hence positive values represent improvement in the score.

and deletions were based. It can be seen from Figure 21 how the probability of a given pronunciation varies with the dialect region against which it is assessed. In the case of the word "19", one of the pronunciations shown (19*3BI: /n ay ng tcl t iy n/) is highly probable across all districts, whereas another of the variants (19*1M: /n ay n v tcl t iy n/) is highly unlikely except in district dr2, hinting that speakers from dr2 may be more inclined to voicing during what should be the silence of the tcl. For the second word, "flann," three variant pronunciations are shown. One (flann*2BB: /zh l ae n/) enjoys universal unpopularity, another (flann*1Z: /f l ey ae n/) is reasonably successful in four of the districts, and the other (flann*3AY: /th l ae n/) is likely

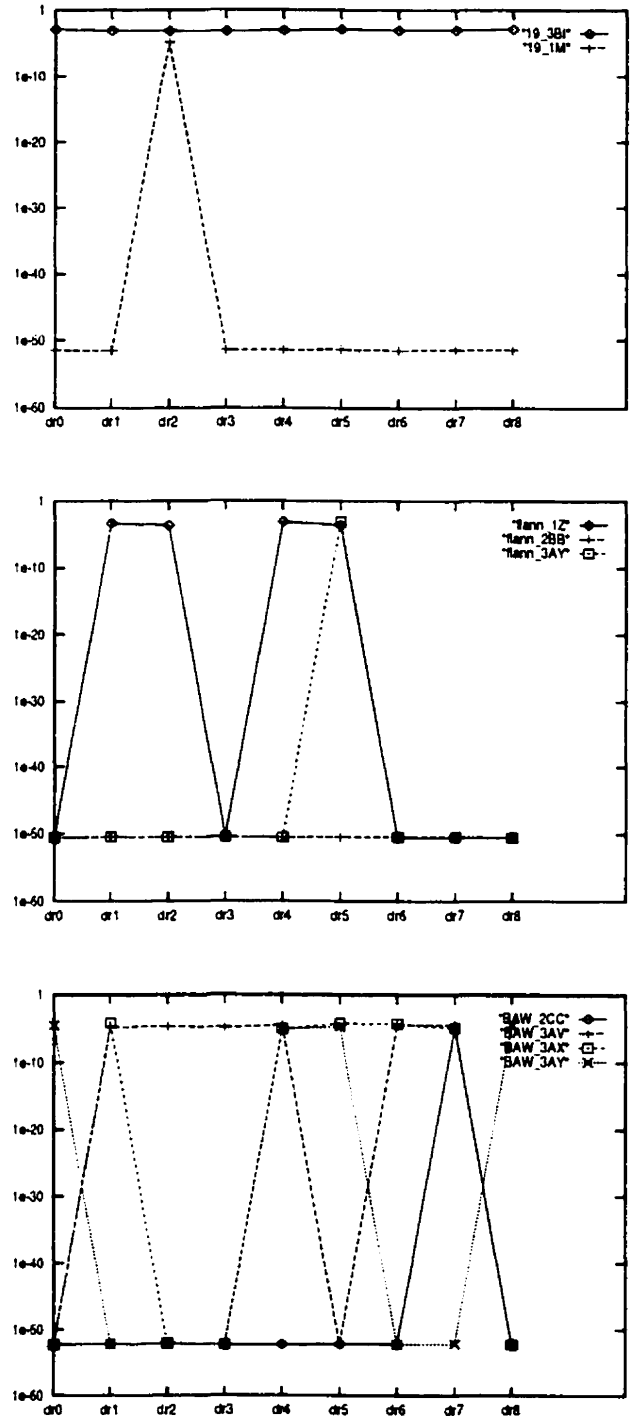

Figure 21 Effect of Dialect Region on Pronunciation Likelihood Shown is the (log) probability of observing a particular pronunciation of three words in the different TIMIT dialect regions (dr0 is an artificial region created by pooling data from the 8 original regions). See text for explanation.

in only one district.

For the last word, "BAW", pronunciations preferred by speaker VP, with heavily accented Bangladeshi English, are plotted. Each of the four variants differs only in one vowel:

BAW*2CC s pcl p iy dcl d bcl b ow dcl d

BAW*3AV s pcl p iy dcl d bcl b aa dcl d

BAW*3AX s pcl p iy dcl d bcl b ah dcl d

BAW*3AY s pcl p iy dcl d bcl b ao dcl d

While BAW*3AV fares better than the others, no one variant can be seen as reliably favoured. This may be more a reflection of the unsuitability of the TIMIT-based statistics, than of how likely native American-English speakers might be to pronounce the word in this fashion.

6.3 Recognition

Once a set of class 3 pronunciations is established, the finite state network for the task grammar is rebuilt with all of the variant pronunciation candidates appearing in parallel. The ensuing recognition provides, if the misrecognition is corrected, the acoustic score for the best of the candidates. Iteration of this procedure, with the previous iteration's 'winner' deleted, provides scores for as many candidates as succeed, or the *N*-best, whichever occurs first.

This recognition step is relatively straightforward. One possible improvement would be the use of context dependent acoustic models in the recognition. Apart from providing outright better recognition, proposed variants would be more refined, though possibly more expensive to generate.

6.4 Preferred Pronunciations

The best pronunciations from the recognition step are preserved in a set of speaker specific preferred pronunciations. The conclusion from the first set of experiments (§ 5.3) is that these pronunciations are not necessarily safe for general consumption. This first set failed due to variant pronunciations trained on a limited and non-representative training set. Indeed, the results for those speakers whose accents are different from the native American-English spoken by the members of the training set bear this out (see Table 13).

The lesson from this first exercise is that a pronunciation developed for one speaker cannot safely be generalized until it is seen to be 'good' for other speakers. With this in mind, a more refined pronunciation dictionary was developed to allow maintaining distinct canonical and acquired pronunciations (see § 6.1). Alternate pronunciations are now acquired by speaker, and post acquisition, may be processed to determine if they capture something more general.

This differs from speaker dependent systems, which also develop pronunciations by speaker, in that the latter typically require users to utter some number of words to train the system before it is used. The pronunciations so acquired are intended for use only by that speaker, without any effort to find ways of generalizing pronunciations for use by other speakers. The system developed in this thesis begins with at least a fully developed canonical PD, and can use, at a stroke, one or more sets of variant pronunciations.

6.5 Generalizing Alternate Pronunciations

The central question posed in this thesis concerns updating the PD automatically. The first set of experiments argued against a very general approach, at least with the

amount of training data available. As a result, a change in approach was made such that each speaker's alternate pronunciations are tied to that speaker. Is it, then, possible to generalize from the pronunciations so acquired, to capitalize on what can be learned from the correction of individual misrecognitions?

One might first look, as a broad test of 'generalizability,' at how well variants from speaker *x* perform on speaker *y*. Table 16 presents some cross-speaker tests (compare with the column for Method 7 in Table 14).

A different way to answer this question is to examine the pronunciations acquired to observe how ubiquitous the corrections are (see Table 17). As can be seen, no pronunciations lend themselves to universal appeal, and few are shared

Table 16 Effect of Mismatching Variants to Speakers The tables show the WER (upper) and SER (lower) when a particular speaker is assessed using variants generated for a speaker other than itself.

Recognition for Speaker	Using Variants from Speaker								
	JA	ZA	JB	MB	DS	QM	VP	TS	VV
JA	—	1.73	0.43	1.73	0.87	1.73	4.33	1.73	1.73
ZA	2.25	—	1.69	2.25	1.69	2.25	2.25	1.68	1.12
JB	5.83	6.67	—	6.25	3.75	6.25	12.92	6.67	11.25
MB	2.70	2.70	2.70	—	2.70	2.70	2.26	2.25	3.15
DS	5.00	4.17	2.92	4.58	—	4.58	10.83	4.17	10.00
QM	6.44	7.30	6.44	5.58	7.30	—	6.01	6.01	6.01
VP	20.00	20.83	21.55	23.75	22.08	21.67	—	24.17	14.17
TS	3.06	2.62	3.49	2.62	1.31	3.06	3.06	—	0.87
VV	13.90	13.37	12.83	12.83	13.37	11.76	9.63	11.76	—

Recognition for Speaker	Using Variants from Speaker								
	JA	ZA	JB	MB	DS	QM	VP	TS	VV
JA	—	8.00	4.00	8.00	4.00	4.00	8.00	8.00	8.00
ZA	14.28	—	14.29	14.29	9.52	14.29	14.29	9.52	9.52
JB	34.62	38.46	—	38.46	23.08	42.31	50.00	38.46	34.62
MB	20.00	20.00	20.00	—	20.00	20.00	16.00	16.00	24.00
DS	30.77	23.08	23.08	26.92	—	26.92	26.92	23.08	26.92
QM	20.00	28.00	20.00	12.00	24.00	—	20.00	16.00	16.00
VP	50.00	53.85	53.85	53.85	50.00	50.00	—	57.69	50.00
TS	24.00	16.00	28.00	24.00	8.00	24.00	32.00	—	8.00
VV	52.38	47.62	47.62	57.14	47.62	52.38	52.38	42.86	—

Table 17 Ubiquitousness of Variant Pronunciations The table shows, for selected misrecognized words, how widespread across different speakers is a particular variant pronunciation's success. The labels appearing in a speaker's column indicate that the pronunciation of the word with that label is in the set of successful variants for this speaker and sentence. The best scoring pronunciation appears first in each list.

	Speaker								
Word	JA	ZA	JB	MB	DS	QM	VP	TS	VV
13							3BX 1R 3J 1B 3N		3J 2L 2BS 3BN 3H 2BO 3BX 2BY 3K 3BL
19		3BI	3M 3BI		3BI		3BI	3BI	
AVA				2BK 2DQ 2CK 2DS 2DL 2DX 2DP	2AD 1U 2V 2AH 2AR 1AQ 2AP 2AK	2BX 2BQ 2C 2BP 2BO 2DJ 2BW	2AP 2BO 2BM 2BN	2BO 2BP 2BM 2BN 2BW 2BR	3D 3CR 3AN 2X 3E 1DO 1CN 2ER 1G 1K

across many speakers. There are, nonetheless, some learned pronunciations that can be promoted to class 2, e.g., 19*3BI.

6.6 Results

The evaluation set results shown in Table 14 argue compellingly in favour of the thesis, with an average reduction in WER of 77.4%, in SER of 72.2%. The results on the test set (Table 15) are not so striking, being 45.3% and 37.7% respectively. Note, too, that neither domain nor speaker adaptation were performed. Further improvement in recognition should be expected by adding the use of these techniques.

One of the first questions raised by the results, even on the evaluation set, is the unevenness of success at correcting misrecognition. What this really demonstrates is that variants derived using data reflecting native American-English speakers work best on native American-English speakers: high rates of correction are shown for speakers JA, JB, QM and TS. Much lower rates of correction are shown for speakers MB, and VP. Speaker JB, without use of the iterative method, would post a WER improvement of only -26.72%, and of -30.00% for SER. This, apart from favourably showing the corrective power of iterative transformation, suggests that something

about JB's speech puts it on a par with a speaker like VP. That something is his rate of speech. Table 10 shows the rate of speech of the nine speakers in the evaluation and test sets, expressed as phonemes/second. The average speaker rate across these nine speakers is 11.5 phonemes/second. This puts both JB (12.9) and DS (12.7) more than one standard deviation above this average. Some of the difficulties posed by fast speech were discussed in § 2.1.2. Is the speech of speakers JB and DS fast enough to incur these errors? Apart from the suggestion in the affirmative shown in the results, it is reported that a phoneme rate as little as one standard deviation above the average is enough to increase error rate [56], and the increase can be dramatic [43]. Interestingly, Siegler and Stern tried introduction of multiple pronunciations to their PD in an attempt to allow for the effects of fast speech on intra-word transitions. They report observing no improvement, but suggest "...it is possible that such an approach could be more successful with a more complete and systematic set of transformation rules." While the substitution and rule-based methods do not appear to have been dramatically effective, iterative transformation did show effective reduction in error rate; study of rules suggested by pronunciations so derived may prove useful in dealing with fast speakers.

There are a few points to consider when interpreting the test set results. The first is that the test set is small (see column 2 in Table 15); in many cases there are fewer than 10 sentences, so a single error is enough to push the string error rate to 10% or more. Secondly, not all of the words appearing in the test set appear in the evaluation set, and so there can be no variants available to correct their misrecognition. This occurs in four sentences affecting four speakers. Thirdly, the training data for the test consists of a relatively small number of sentences. In the one case where it is possible, that of speaker JA, the introduction of additional training material (on the original 47 sentence training set) eliminates the one word misrecognized in his test sentences. Were one to discount sentences with words not present in the evaluation set, and allow JA's additional training data, the test set results are improved (see

Table 18). Alternatively, if one performs training on the novel words as the appear in the test sentences, the results are also improved.

6.7 Future Work

The current system for developing and using alternate pronunciations has several parameters that may be adjusted. A great deal of work can be done exploring what

Table 18 Improved Test Set Results The table shows how the test set results reported in Table 15 are affected by removal novel words, by using additional training material, and by use of variants trained on the novel words appearing in the test set. In the upper table (WER), the number of words originally and following deletion of novel words is shown. In the lower table (SER), the number of sentences originally and following deletion of sentences containing novel words is shown. Error rates shown are those obtained by beginning with the lowest error rate provided by any of the methods reported earlier, and applying the indicated operation. Canonical error rates are reproduced here from Table 10 for convenience.

Speaker	Words		Base	Delete	Use Extra	With Variants
Id	Original	After	Canonical	New Wds	Train Based	of New Wds
JA	103	97	0.97	1.03	0.00	0.97
ZA	65	63	4.62	1.59	—	1.54
JB	103	97	10.68	10.31	—	8.74
MB	81	79	1.23	0.00	—	0.00
DS	87	84	5.75	4.76	—	4.60
QM	72	70	2.78	0.00	—	0.00
VP	87	84	18.39	7.14	—	6.90
TS	87	84	4.60	0.00	—	1.15
VV	76	73	17.11	5.48	—	5.26
Average	84	81	7.35	3.37	0.00	3.24

Speaker	Sentences		Base	Delete	Use Extra	With Variants
Id	Original	After	Canonical	New Wds	Train Based	of New Wds
JA	11	7	9.09	14.29	0.00	9.09
ZA	8	6	25.00	16.67	—	12.50
JB	11	7	45.45	28.57	—	27.27
MB	9	7	11.11	0.00	—	0.00
DS	10	7	30.00	28.57	—	20.00
QM	9	6	11.11	0.00	—	0.00
VP	10	7	70.00	28.57	—	30.00
TS	10	7	20.00	0.00	—	10.00
VV	9	6	44.44	33.33	—	22.22
Average	9	7	29.58	16.67	0.00	14.56

parameter settings work best overall, for a collection of speakers, and/or for an individual speaker.

It has become nearly axiomatic in speech recognition to argue that more training data is required†. Certainly, there is much that can be done in developing variant pronunciations for speakers that can be grouped into accent- or dialect-based sets. Ideally, one can foresee constructing 'standard' sets of alternate pronunciations for specific groups, e.g,. Bangladeshi-accented English speakers. A recognition system with some ability to identify a speaker's accent would be able to instantiate such a set, dynamically, during recognition when used by a speaker 'from' that set. These sets could even be used by other recognition systems, so long as they used a compatible representation for word pronunciations. This would require the use of a large amount of clearly segregated data, some to serve as fodder for the recognizer so as to develop alternate pronunciations, and some to be used in developing pronunciation statistics for use in assessing the likelihood of pronunciations.

Another aspect of the system which would benefit from further exploration is the set of rules distilled from pronunciations and used as the basis of the rule-based variant generator. To what extent may specific detail be dropped from this set, i.e., what is the tradeoff between making a smaller, more general rule set, and power at generating useful variant pronunciation candidates?

There are performance issues that can be explored to make the acquisition and deployment of alternate pronunciations fast enough to be used in a practical way during live recognition. The current system, constructed as a research work-bench, performs its variant generation and testing off-line. The former could be performed easily enough in faster-than-real-time, and the testing sped up by making use of the search space already built during the (mis)recognition. One would splice into the

† As eloquently expressed by L. Rabiner, "There's no data like more data"

space the parallel collection of variants, and run the recognition forward from the state immediately prior to the point of insertion.

Beyond these issues, in the case of pronunciations that are found to be successful — for an individual speaker or for some collection of speakers — a reasonable question to ask is: do these pronunciations have any generalizable quality? That is, can what is learned about correcting one misrecognition be used to correct others, whether from the same speaker or another? Does the corrective power generalize so as to allow other words having the same or similar contexts to be 'pre-emptively' corrected, before they are misrecognized?

In addition to all of the above, a pressing question is how well these techniques scale to use in large vocabulary systems: tasks having tens of thousands of words and no grammar so constraining as the finite state network used for the comparatively straightforward air traffic control task investigated here. A promising direction to follow is the use of the methods described here in improving the word candidate set size in a word lattice for large vocabulary tasks.

6.8 Conclusion

Automatic speech recognition has advanced a great deal in the last 20 years. Benefiting from almost unimaginable improvements in processor speed and memory capacity, as well as intensive research work on representation and search problems, CSR systems today are at the threshold of wide-spread, general use. Many systems are already deployed productively in settings where they are used by one (or a small number) of speakers in highly circumscribed applications. The next major leap will be to systems that can perform continuous word recognition of ever less unconstrained speech. A necessary step leading to this leap will be the ability to handle widely varying pronunciations for words from the large user community envisaged.

This thesis takes precise aim at this necessary step, investigating the question of whether a pronunciation dictionary can be updated automatically so as to improve recognition accuracy. From investigation of quite simple techniques applied to a small, but highly diversified, set of speakers, the conclusion is that this can, in fact, be done. It can be done in such a way that alternate pronunciations can be learned and made available immediately for future recognition, without resorting to extensive and computationally expensive model retraining. Further, it suggests that much greater improvement than was possible to achieve with the data available may be attained with development of sets of accent- or dialect- specific alternate pronunciations.

These qualities were required for the air traffic control application explored in the thesis research. The setting for this application is a training facility which may be installed anywhere in the world, and so encounter users speaking a language which is foreign to most of them, English. The features which make this system successful for this application make it suitable for many others.

The system as currently developed provides many opportunities for future research and development work which can improve its corrective power and performance.

Appendix 1: Speech Recognition Tasks and Corpora

A1.1 ATS2 Task and Corpus

The ATS2 task was developed with Loral Federal Systems Company (FSC), of Rockville, MD., during the winter of 1994/1995. The application is an ATC simulation of approach scenarios for JFK International airport in New York.

Sentences in this task are entirely determined by a grammar (see Figure 22). The task has 116 words and a perplexity of 3.3. Words in the task are modeled as concatenations of TIMIT phoneme units.

The ATS2 corpus consists of two distinct sets of recordings. In both cases, recordings were made at Loral of read speech from a small number of speakers. As not all of the available speakers were 'fluent ATC speakers', sentences to be read were generated from the grammar. The prime consideration in generating the sentences was to provide broad coverage of the grammar.

The first set of recordings (September, 1994) consists of three speakers reading 47 sentences each. Recordings were made using a Shure DY-10 headset mounted, close-talking microphone, sampled directly by a Gradient Technologies DeskLab 216 A/D converter at 16 kHz, and represented as 16 bit (linear) signed integers. The recordings were made in an active machine room, with a high content of background white noise, but little of this noise is detectable in the recordings. All speakers were North American native English speakers, but of very different dialects.

The second set of recordings (January, 1995) consists of nine speakers reading 37 sentences each. Recordings were made using the same Shure DY-10 microphone, under the same conditions, sampled at 48 kHz by DAT recorder, and later downsampled to 16 kHz. These speakers provide a variety of accents including: Bangladeshi, Québec French (an employee of ATS Aerospace), Indian, Arab/British, and several

native North American English dialects.

The total collection of sentences was divided into three sets for the work conducted here:

Sentence Numbers	Num. of Sentences	Num. of Speakers	Total num. of Sentences	Disposition
01 - 47	47	3	138	training
70 - 95	26	9	228	evaluation
96 - 106	11	9	99	test

Some of the sentences are 'misread' with respect to the prompting text; of these, those that are still grammatical are retained, the others discarded, thus the total number of sentences does not correspond, in all cases, to the product of sentences × speakers. No sentences numbered 48 to 69 were recorded.

```
CALL_SIGN =     AAL     { <110 - 119> | <210 - 219> }
                ACA     { <110 - 119> | <910 - 919> }
                AMX     { <310 - 319> | <410 - 419> }
                AVA     { <510 - 519> | <610 - 619> }
                BAW     { <710 - 719> | <810 - 819> }
                COA     { <210 - 219> | <310 - 319> }
                DAL     { <410 - 419> | <510 - 519> }
                EIN     { <310 - 319> | <410 - 419> }
                JAL     { <610 - 619> | <710 - 719> }
                KAL     { <810 - 819> | <910 - 919> }
                NWA     { <110 - 119> | <210 - 219> }
                SAB     { <510 - 519> | <610 - 619> }
                UAL     { <710 - 719> | <810 - 819> }
                USA     { <110 - 119> | <910 - 919> }
                VRG     { <210 - 219> | <310 - 319> }
                5 2 lima papa
                1 0 echo charlie
                2 6 3 9 x-ray
                papa juliet quebec alpha
                2 1 0 charlie lima
                6 1 7 zulu
                1 0 1 5 delta
                X-ray X-ray lima quebec
                6 6 2 mike golf

COMMAND =       { descend | climb } { to | and maintain } flightlevel FLIGHTLEVEL [ altimeter <0 - 2><0 - 9><0 - 9><0 - 2
                { fly | maintain } heading HEADING [ degrees ] [ vector for New York approach ]
                turn { left | right } [ heading ] HEADING [ degrees ]
                { increase | reduce } speed [ to ] SPEED [ knots ] [ indicated [ for sequencing to runway 0 4 { left | righ
                cleared for { ils | visual } 0 4 { right | left } approach
                proceed [ directly ] to FIX [ on the HEADING [ degree radial ] at <1-9><0-9><0-9> dme ]
                resume own navigation via [ FIX | ROUTE ]
                affirmative [ flightlevel ] FLIGHTLEVEL [ ident ]
                report { [ reaching ] [ flightlevel ] FLIGHTLEVEL } | { [ the ] outer marker inbound }
                contact AIRPORT on 1 2 <0,3-4> point 9
                traffic <1-12> oclock <0-10> miles [ BOUND ] TYPE at [ flightlevel ] FLIGHTLEVEL
                roger contact tower 1 2 <0,3-4> point 9
                ident

FLIGHTLEVEL= { { 1 <5-9> | <2-9><0-9> | <1-5><0-9><0-9> | 6 0 0} | { <1-9> | <1-5><0-9> | 6 0  thousand } }

SPEED= { <6-9><0-9> | <1-4><0-9><0-9> }

HEADING= <0-359>

BOUND= { north [ east | west ] | east | south [ east | west ] | west } bound

TYPE=   airbus
        boeing 7 { oh | 20 | 30 | 40 | 50 } 7
        bac 1 11
        citation
        d c { 8 | 10}
        king air
        gulfstream
        l 10 11
        lear jet
        twin commanche
        m d 80
        westwind

ROUTE=  jet 1 7 4
        alpha 5 2 3
        jet 6 0

FIX=    solberg
        wavey
        bermuda
        champ
        east_texas
        shipp
        carmel
        bergh
        dixie
        j_f_k
        modena
        linnd
        camrn
        coyle
        flann
        daner

AIRPORT=    New York center
            New York approach
```

Figure 22 ATS2 Task Grammar

The prompting texts of the sentences appear below:

01 American 1 11 descend and maintain flightlevel 1 9 0
02 Air Canada 1 12 fly heading 1 5 0 degrees
03 Air lingus 3 13 turn left 2 4 0 degrees
04 Air Mexico 3 14 increase speed 1 8 0 knots
05 Avianca 5 15 cleared for I L S 0 4 left approach
06 Continental 2 16 resume own navigation via Solberg
07 Delta 4 17 affirmative 5 thousand
08 Japan Airlines 6 18 report reaching flightlevel 2 9 0
09 Northwest 1 19 contact New York Center on 1 2 0 . 9
10 Sabena 5 11 traffic 12 o'clock 5 miles northbound Airbus at 1 0 thousand
11 Speed Bird 7 12 roger contact tower 1 2 3 . 9
12 U S Air 1 13 climb to flightlevel 3 5 0 altimeter 2 9 9 2
13 United 7 14 maintain heading 0 4 0
14 Varig 2 15 turn right heading 1 4 5
15 Korean Airlines 8 16 reduce speed 2 2 0
16 American 9 17 cleared for visual 0 4 left approach
17 Air Canada 9 18 resume own navigation via J 1 7 4
18 Air lingus 4 19 affirmative flightlevel 1 8 0
19 Air Mexico 4 11 report reaching flightlevel 1 8 0
20 Avianca 6 12 contact New York approach on 1 2 4 . 9
21 Continental 3 13 traffic 7 o'clock 10 miles southbound Boeing 7 57 at 2 7 0
22 Delta 5 14 roger contact tower 1 2 0 . 9
23 Japan Airlines 7 15 descend and maintain 1 1 thousand
24 Northwest 2 16 fly heading 1 0 0 vector for New York approach
25 Sabena 6 17 turn left heading 2 0 5 degrees
26 Speed Bird 8 18 increase speed 3 2 0
27 U S Air 9 19 cleared for I L S 0 4 right approach
28 United 8 11 resume own navigation via Wavey
29 Varig 3 12 affirmative flightlevel 3 9 0 ident
30 Korean Airlines 9 13 report flightlevel 4 1 0
31 American Airlines 1 14 contact New York Center on 1 2 3 . 9
32 Air Canada 1 15 traffic 8 o'clock 9 miles eastbound M D 80 at flightlevel 3 3 0
33 Air Lingus 3 16 roger contact tower 1 2 4 . 9
34 Air Mexico 4 17 resume own navigation via Bermuda
35 Avianca 5 18 resume own navigation via Champ
36 Continental 2 19 resume own navigation via East Texas
37 Delta 4 11 resume own navigation via Carmel
38 Japan Airlines 6 12 resume own navigation via Bergh
39 Northwest 1 13 resume own navigation via Dixie
40 Sabena 5 14 resume own navigation via J F K
41 Speed Bird 7 15 resume own navigation via Modena
42 U S Air 1 16 resume own navigation via Linnd
43 United 7 17 resume own navigation via Camrn
44 Varig 2 18 resume own navigation via Coyle
45 Korean Airlines 8 19 resume own navigation via Flann
46 American 9 11 resume own navigation via Daner
47 Air Lingus 4 12 resume own navigation via J 6 0
70 American 1 10 descend and maintain flightlevel 2 2
71 Air Canada 9 12 fly heading 1 7 0
72 Air Mexico 3 14 turn right 0 8 0 degrees
73 Avianca 5 16 increase speed 2 5 0
74 Speed Bird 7 18 cleared for I L S 0 4 left approach
75 Continental 3 11 proceed to Wavey
76 Delta 4 13 resume own navigation via Flann
77 Air Lingus 3 15 affirmative 1 8 0
78 Japan Airlines 6 17 report reaching 2 3 0

79 Korean Airlines 8 19 contact New York approach on 1 2 0 . 9
80 Northwest 1 12 traffic 2 o'clock 7 miles northbound D C 8 at 1 4
81 Sabena 5 14 roger contact tower 1 2 0 . 9
82 United 7 16 ident
83 U S Air 1 18 climb and maintain 2 7 thousand
84 Varig 2 11 maintain heading 3 3 0 degrees
85 American 2 13 turn left heading 1 4 0
86 Air Canada 9 15 reduce speed 1 6 0
87 Air Mexico 4 17 cleared for visual 0 4 right approach
88 Avianca 6 19 proceed directly to Dixie on the 2 9 0 degree radial at 1 0 9 D M E
89 Speed Bird 8 12 resume own navigation via jet 60
90 Continental 3 16 affirmative flightlevel 4 2 0
91 Delta 5 18 report reaching the outer marker inbound
92 Air Lingus 4 11 contact New York Center on 1 2 3 . 9
93 Japan Airlines 7 13 traffic 9 o'clock 3 miles southbound Airbus at 1 6
94 Korean Airlines 9 17 roger contact tower 1 2 3 . 9
95 Northwest 2 19 ident
96 Sabena 6 12 descend and maintain flightlevel 1 2 0 altimeter 2 9 9 2
97 United 8 14 fly heading 0 2 0
98 U S Air 9 16 turn right 2 1 0 degrees
99 Varig 3 18 increase speed 6 5
100 American 1 11 cleared for I L S 0 4 right approach
101 Air Canada 1 13 proceed to Solberg
102 Air Mexico 3 17 resume own navigation via Bermuda
103 Avianca 5 19 affirmative 4 2 thousand
104 Speed Bird 7 12 report 1 7 0
105 Continental 2 14 contact New York approach on 1 2 4 . 9
106 Delta 4 16 traffic 4 o'clock 1 mile eastbound Boeing 7 57 at 2 2

A1.2 WSJ Task and Corpus

The Wall Street Journal task was begun in the early 1990s as an effort to provide a large, multi-purpose corpus to the speech research community: (initially) 400 hours of speech and a corresponding large text corpus (47 million words) for training language models. The task featured natural language with high perplexity. Data were organized into at least two levels of vocabulary size, 5,000 and 20,000 word. The speech was collected under clean recording conditions using, simultaneously, a Sennheiser HMD414 close-talking microphone and a secondary microphone (of varying type). Data was collected by MIT, SRI and Texas Instruments. The speech is read, by equal numbers of male and female speakers, from articles appearing in the Wall Street Journal newspaper. Speakers were "...chosen for diversity of voice quality and dialect" [44].

Care was taken with organizing the data so as to provide material for training and testing of speaker dependent and speaker independent recognizers, as well as the providing of extra training material — "phonetically rich" sentences — for the development of speaker adaptive systems.

The corpora are released through the Linguistic Data Consortium (LDC) on CD-ROMS.

A1.3 TIMIT

In the mid-1980s Texas Instruments and MIT collaborated to produce what has become, in all likelihood, the most used corpus in the speech recognition community, TIMIT. The problem addressed by TIMIT was the lack of consistent, abundant, phonetically labeled data suitable for training acoustic models. The solution was to obtain a set of high quality recordings of read speech and provide the phonetic segmentation of each utterance. The text of the sentences was chosen so as to be "...phonetically rich. Care was taken to have as complete a coverage of left- and right-context for each phone as possible" [55]. A further objective of some of the sentences was to afford an opportunity to observe "...dialectical and phonological variations across speakers."

All told, each of 630 speakers read 10 sentences. In an early status report the ratio of male to female speakers is reported as 2:1 [18]. The speakers were categorized as being from one of eight different 'dialectic regions:'

Area Designation	Area Location
1	New England
2	Northern
3	North Midland
4	South Midland
5	Southern
6	New York City
7	Western
8	Army Brat

While this categorization may have been somewhat *ad hoc*, it does result in measurably different pronunciation probabilities between dialectic regions (see, e.g., Figure 21)

Since the objective of the set of sentences was primarily to provide good coverage of phonemes for training of acoustic models, including in-context occurrences, it is debatable how well the word sequences in the corpus reflect 'normal speech.' Consequently, a bigram language model trained on TIMIT sentences may not be a good representation of 'normal' conversational speech. Nonetheless, much of the work reported in this thesis uses just such a bigram language model for 'raw' phoneme recognition (see, e.g., Figure 19).

Another consideration of TIMIT as a corpus for training acoustic models is that it provides a richer set of acoustically labeled units than some other recognition corpora use, e.g., WSJ. Table 19 shows the full set of phonemes used in the TIMIT corpus. The ATS2 task uses all of these units except /axr, ax-h, nx, em, eng, ux/.

Vowels

eh	ɛ	ih	ɪ
ao	ɔ	ae	æ
aa	ɑ	ah	ʌ
uw	u	uh	ʊ
er	ɝ	ux	ü
ay	$ɑ^{y}$	oy	$ɔ^{y}$ (ɔy)
ey	e^{y} (e)	iy	i^{y} (i)
aw	$ɑ^{w}$	ow	o^{w} (o)
ax	ə	axr	ɚ
ix	ɨ	ax-h	ə̥

Fricatives

s	s	sh	š (ʃ)
z	z	zh	ž (ʒ)
ch	č (tʃ)	jh	ǰ (dʒ)
th	θ	dh	ð
f	f	v	v
h	h		

Liquids, Glides

l	l	el	l̩
r	r	w	w
y	y		

Nasals

m	m	em	m̩
n	n	en	n̩
ng	ŋ	eng	ŋ̩
nx	ɾ̃		

Stops

pcl	p°	bcl	b°
p	p	b	b
tcl	t°	dcl	d°
t	t	d	d
kcl	k°	gcl	g°
k	k	g	g

Table 19 **TIMIT phoneme units** Shown are the phoneme labels used in the TIMIT corpus, and their corresponding symbols as in common use among linguists (with IPA symbols in parentheses where they differ). See text for details (also Table 2).

Appendix 2: The Roger Speech Recognizer

The recognition system used for all experimental work reported in this thesis was developed over several years in the Speech Laboratory at McGill University's School of Computer Science (part of the Center for Intelligent Machines, CIM). The recognizer was built by several individuals (including the author) over this period. The system, called ROGER, is a classical HMM based recognition engine, fed by an equally classical feature extractor.

A2.1 System Architecture

ROGER is implemented entirely in software, using no special purpose hardware (save the A/D converter). It is written in C to run under the UNIX operating system, and currently runs on Sun, Hewlett-Packard, IBM RISC System 6000 and (at least Intel-based) Linux platforms.

The main recognition component of ROGER consists of three processes which communicate using two virtual channels (control, data) implemented over UNIX pipes. This design was chosen for ease of use in a research setting; a more practically-oriented implementation might choose more efficient means for inter-process communication.

A2.2 System Components

A2.2.1 Recorder

This component performs data acquisition and acts as a crude command interface and file-manager for the overall recognizer. In that it is the only component that has

to deal directly with the host computer's hardware, it is the component requiring the most effort in porting ROGER to a new platform, given that no standard (i.e., multi-platform) interface has yet emerged for application interaction with audio hardware.

The recorder places onto the data channel sampled audio to be presented to the recognizer. This audio may be from a previously stored file, or 'live' input from microphone or line-level sources. The sampling rate is currently fixed at 16 kHz, and only 16 bit linear (monaural) data are supported.

The recorder places onto the command channel tags for use by downstream processes (e.g., 'start-of-utterance', 'end-of-utterance', etc.). A more elaborate graphical command interface is available, though it is nothing more than 'window dressing', i.e., a graphical layer wrapped around the basic recognition engine. It receives output from the recognizer component for display, as well as keyboard input from users, and sends its output to the recorder component as input. The graphical interface was not used during the experiments reported in the thesis.

A2.2.2 Feature Extractor

Sampled audio data is blocked into frames of 20 ms. duration (320 samples). To these data is applied a Hamming window. Successive applications of the window overlap by 50%, i.e., there is new windowed data for every 10 ms. of audio data.

Computation of features is performed in both the time and frequency domain. The time domain features are the gross energy per window, computed as the sum of the squared values in the window, and normalized with respect to the peak energy of the utterance. As ROGER aims to perform live recognition, it is not possible to wait until the entire utterance has been heard to determine the peak energy against which to normalize. Instead, ROGER uses a guess at what the peak energy is likely to be. In addition to the overall energy E, the first derivative with respect to time , ΔE_i for

window i is computed simply as the difference $E_i - E_{i-1}$.

In the frequency domain, 12 mel cepstral coefficients (see § 2.2.2) are computed. Here, too, the first derivative with respect to time is provided, derived for window i and coefficient j as:

$$\frac{d}{dt} c_{ij} = \sum_{k=1}^{m} \frac{c_{ij+k} - c_{ij-k}}{k^2}$$

where m is the width of window used in computing the difference.

From these computations, a feature vector of 26 elements is produced for each window, and serves as the actual data modeled by the recognizer's acoustic models.

A2.2.3 Recognizer

The recognizer uses continuous Gaussian distributions for both symbol emission and transition probabilities. The models were trained on the TIMIT corpus. The topology of these context independent models provides for a greater number of transitions to appear near the beginning and end of a model, with comparatively few internally, as a way to model better the greater variation experienced at the 'extremities' of a phoneme (i.e., as it is used in different contexts).

The recognizer uses a stochastic finite state grammar. Auxiliary utilities supplied with ROGER include a program for compiling grammars into the form required by the recognizer. The network for the ATS2 task, for example, contains 5,606 states and 7,519 transitions.

Recognition behaviour may be controlled with several parameters, including adjustment of the beam threshold used in the Viterbi search, and whether the constraint that a search path must end in a final state of the grammar in order to be hypothesized is enforced or not. In experiments reported in the thesis, partial paths were allowed (hence potentially reducing word error rates but not impacting string

error rates).

References

The following abbreviations appear in the listings of publications:

AAAI	American Association for Artificial Intelligence
ACL	Association for Computational Linguistics
ICASSP	International Conference on Acoustics, Speech, and Signal Processing
ICSLP	International Conference on Spoken Language Processing
Proc.	Proceedings of

References

1. Akmajian, A., Demers, R. A., Farmer, A. K., and Harnish, R. M., *Linguistics: An Introduction to Language and Communication*, MIT Press (1990).
2. Bahl, L., Das, S., de Souza, P., Epstein, M., Mercer, R., Merialdo, B., Nahamoo, D., Picheny, M., and Powell, J., "Automatic Phonetic Baseform Determination," *Proc. ICASSP*, pp. 173 - 176 (1991).
3. Bahl, L. R., Brown, P. F., de Souza, P. V., Nahamoo, D., and Mercer, R. L., "Maximum Mutual Information Estimation of Hidden Markov Models Parameters for Speech Recognition," *Proc. ICASSP*, pp. 49-52 (1986).
4. Baker, J., "Stochastic Modeling for Automatic Speech Understanding," *Speech Recognition*, pp. 521 - 541 (1975).
5. Baum, L. E. and Egon, J. A., "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology," *Bulletin of the American Meteorological Society* **73** pp. 360 - 363 (1967).
6. Berkling, K. M and Barnard, E., "Language Identification of Six Languages Based on a Common Set of Broad Phonemes," *Proc. ICSLP*, pp. 1891 - 1895 (1994).
7. Bloothooft, G. and Plomp, R., "Spectral analysis of sung vowels. II: The effect of fundamental frequency on vowel spectra," *JASA* **77**(4) pp. 1580-1588 (1985).
8. Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Wadsworth Inc. (1984).

9. Brown, P. F., de Souza, P. V., Mercer, R. L., and Picheny, M. A., "A Method for the Construction of Acoustic Markov Models for Words," *IEEE Transactions on Speech and Audio Processsing* **1**(4)(1993).

10. Byrne, B., Finke, M., Khudanpur, S., McDonough, J., Nock, H., Riley, M., Saraclar, M., Wooters, C., and Zavaliagkos, G., "Pronunciation Modelling for Conversational Speech Recognition: A Status Report From WS97," *Proc. of the Fifth LVCSR Summer Workshop*, pp. 26 - 33 (1997).

11. Church, K., "Phonological parsing and lexical retrieval," pp. 53-69 in *Spoken Word Recognition*, ed. Fraudenfelder, U. and Komisarjevsky Tyler, L.,MIT Press (1987).

12. CMU, *The Carnegie Mellon Pronouncing Dictionary*, Carnegie Mellon University (1993).

13. Cocker, C., "A Dictionary-Intensive Letter-to-Sound Program," *Acoustical Society of America* **78 Suppl. 1** p. S7 (1985).

14. Cohen, M., Rivlin, Z., and Bratt, H., "Speech Recognition in the ATIS Domain Using Multiple Knowledge Sources," *Proc. ARPA Spoken Language Systems Technology Workshop*, pp. 257 - 260 (1995).

15. Cohen, P. and Mercer, P., "The Phonological Component of an Automatic Speech Recognition System," pp. 275 - 320 in *Speech Recognition*, ed. Raj Reddy, (1975).

16. Crystal, T. H. and House, A. S., "Segmental durations in connected speech signals: Preliminary results," *JASA* **72**(3) pp. 705-716 (1982).

17. Ephraim, Y., Dembo, A., and Rabiner, L. R., "A Minimum Discrimination Information Approach for Hidden Markov Modeling," *IEEE Trans. Information Theory* **35**(5) pp. 1001-1013 (1989).

18. Fisher, W. M., Doddington, G. R., and Goudie-Marshall, K. M., "The DARPA Speech Recognition Research Database: Specifications and Status," *Proc. DARPA Speech Recognition Workshop*, pp. 93 - 99 (1986).

19. Gauvain, J. L., Lamel, L. F., Adda, G., and Adda-Decker, M., "Speaker-Independent Continuous Speech Dictation," *Proc. EuroSpeech*, (1993).

20. Gay, T., "Effect of speaking rate on vowel formant movements," *JASA* **63**(1) pp. 223-230 (1978).

21. Gray, R., "Vector Quantization," in *Readings in Speech Recognition*, ed. Waibel, A. and Lee, K. F.,Morgan Kaufmann (1984).

22. Hansen, J. and Arslan, L., "Foreign Accent Classification Using Source Generator Based Prosodic Features," *Proc. ICASSP*, pp. 836 - 839 (1995).

23. Hempell, Godfrey, and Doddington, "The ATIS Spoken Language Systems Pilot Corpus," *Proc. DARPA Speech and Natural Language Workshop*, (June 1990).

24. Hopcroft, J. and Ullman, J., *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (1979).

25. Imai, T., Ando, A., and Miyasaka, E., "A New Method for Automatic Generation of Speaker-Dependent Phonological Rules," *Proc. ICASSP*, pp. 864 - 867 (1995).

26. Jelinek, F., "Self-Organized Language Modeling for Speech Recognition," pp. 450-506 in *Readings in Speech Recognition*, ed. Waibel, A. and Lee, K. F.,Morgan Kaufmann (1990).

27. Junqua, J-C. and Anglade, Y, "Acoustic and Perceptual Studies of Lombard Speech: Application to Isolated Words Automatic Speech Recognition," *Proc. ICASSP*, pp. 841-844 (1990).

28. Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Fosler, E., and Morgan, N., "The Berkeley Restaurant Project," *Proc. ICSLP*, pp. 2139 - 2142 (1994).

29. Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., and Morgan, N., "Integrating Experimental Models of Syntax, Phonology, and Accent/Dialect in a Speech Recognizer," *Proc. AAAI Workshop on Integration of Natural Language and Speech Processing*, (1994).

30. Kipp, A., Wolfertstetter, F., and Ruske, G., "Automatic Generation of Rules for Typical Pronunciation Variants," *Pers. comm.*, (1995).

31. Lamel, L. F., Kassel, R. H., and Seneff, S., "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," *Proc. DARPA Speech Recognition Workshop*, pp. 100 - 109 (1986).

32. Lee, K. F., "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System," *CMU-CS-88-148*, Computer Science Dept., Carnegie Mellon University, (1988).

33. Levelt, W., *Speaking From Intention to Articulation*, MIT Press (1989).

34. Lippman, R. P., Martin, E. A., and Paul, D. B., "Multi-Style Training for Robust Isolated-Word Speech Recognition," *Proc. ICASSP*, pp. 705-708 (1987).

35. Lucassen, J. and Mercer, R., "An Information Theoretic Approach to the Autonomous Determination of Phonemic Baseforms," *Proc. ICASSP*, pp. 42.5.1 - 42.5.4 (1984).

36. Mirghafori, N., Fosler, E., and Morgan, N., "Towards Robustnes to Fast Speech in ASR," *Proc. ICASSP*, pp. 335 - 338 (1996).

37. Møller, A. R., *Auditory Physiology*, Academic Press (1983).

38. Moreno, P. J., Bhiksha, R., and Stern, R. M., "A Vector Taylor Series Approach for Environment-Independent Speech Recognition," *Proc. ICASSP*, pp. 733 - 736 (1996).

39. De Mori, R., Galler, M., Snow, C., and Kuhn, R., "Speech Recognition and Understanding," pp. 125 - 162 in *Approaches to Telecommunications and Network Management*, ed. Liebowitz, J.,IOS Press (1994).

40. De Mori, R., Snow, C., and Galler, M., "On the use of Stochastic Inference Networks for Representing Multiple Word Pronunciations," *Proc. ICASSP*, pp. 568 - 571 (1995).

41. Ney, H. and Noll, A., "Phoneme Modeling Using Continuous Mixture Densities," *Proc. ICASSP*, pp. 437-440 (1988).

42. O'Shaughnessy, Douglas, *Speech Communication*, Addison-Wesley (1987).

43. Pallett, D. S., Fiscus, J. G., Fisher, W. M., Garofolo, J. S., Lund, B. A., and Przybocki, M. A., "1993 WSJ-CSR Benchmark Test Results," *Proc. ARPA Spoken Language Systems Technology Workshop*, (1994).

44. Paul, D. and Baker, J., "The Design for the Wall Street Jouranl-based CSR Corpus," *Proc. ICSLP*, pp. 899 - 902 (1992).

45. Paul, D. B., "Training of HMM Recognizers by Simulated Annealing," *Proc. ICASSP*, pp. 13-16 (1985).

46. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kauffman (1988).

47. Pellom, B. L. and Hansen, J. H. L., "Text-Directed Speech Enhancement Using Phonemic Classification and Feature Map Constrained Vector Quantization,"

Proc. ICASSP, pp. 645 - 648 (1996).

48. Philips, M., Glass, J., Polifroni, J., and Zue, V., "Collection and Analyses of WSJ-CSR Corpus at MIT," *Proc. ICSLP*, pp. 907 - 910 (1992).
49. Picone, Joseph W., "Signal Modeling Techniques in Speech Recognition," *Proc. of the IEEE* **81**(9) pp. 1215-1247 (1993).
50. Port, R. F., "Linguistic timing factors in combination," *JASA* **69**(1) pp. 262-274 (1981).
51. Rabiner, L., Lee, C., Juang, B., and Wilpon, J., "HMM Clustering for Connected Word Recognition," *Proc. ICASSP*, pp. 405 - 408 (1989).
52. Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE* **77** pp. 257-286 (1989).
53. Riley, M., "A Statistical Model for Generating Pronunciation Networks," *Proc. ICASSP*, pp. 737 - 740 (1991).
54. Sataloff, R., "The Human Voice," *Scientific American* **267**(6) pp. 108-125 (1992).
55. Seneff, S. and Zue, V., *Transcription and Alignment of the TIMIT Database*, NIST (1988).
56. Siegler, M. A. and Stern, R. M., "On the Effects of Speech Rate in Large Vocabulary Speech Recognition Systems," *Proc. ICASSP*, pp. 612 - 615 (1995).
57. Sloboda, T., "Dictionary Learning: Performance Through Consistency," *Proc. ICASSP*, pp. 453 - 456 (1995).
58. Snow, C., "Multiple Pronunciations in Multilingual Speech Recognition," *Proc. 3rd CRIM-FORWISS Workshop*, pp. 66 - 74 (1996).
59. Stolcke, A. and Omohundro, S., "Best-First Model Merging for Hidden Markov Model Induction," TR-94-003, International Computer Science Institute (1994).
60. Tajchman, G., Fosler, E., and Jurafsky, D., "Building Multiple Pronunciation Models for Novel Words Using Exploratory Computational Phonology," *Proc. Eurospeech*, (1995).
61. Tajchman, G., Jurafsky, D., and Fosler, E., "Learning Phonological Rule Probabilities from Speech Corpora with Exploratory Computational Phonology," *Proc. ACL*, (1995).
62. Tuller, B., Harris, K. S., and Kelso, J. A. S., "Stress and rate: differential transforms of articulation," *JASA* **71**(6) pp. 1534-1543 (1978).
63. Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Information Theory* **13**(2)(1967).
64. Wooters, C., "Lexical Modeling in a Speaker Independent Speech Understanding System," TR-93-068, International Computer Science Institute (1993).
65. Wooters, C. and Stolcke, A., "Multiple-Pronunciation Lexical Modeling in a Speaker Independent Speech Understanding System," *Proc. ICSLP*, pp. 1363 - 1366 (1994).
66. Zue, V. and Cole, R., "Spoken Language Input," in *Survey of the State of the Art in Human Language Technology*, ed. Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V.,Center for Spoken Langauge Understanding, Oregon Graduate Institute (1995).