INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Lexical Modeling in a Speaker Independent Speech Understanding System

Charles Clayton Wooters

TR-93-068

November 1993

Abstract

Over the past 40 years, significant progress has been made in the fields of speech recognition and speech understanding. Current state-of-the-art speech recognition systems are capable of achieving word-level accuracies of 90% to 95% on continuous speech recognition tasks using 5000 words. Even larger systems, capable of recognizing 20,000 words are just now being developed. Speech understanding systems have recently been developed that perform fairly well within a restricted domain.

While the size and performance of modern speech recognition and understanding systems are impressive, it is evident to anyone who has used these systems that the technology is primitive compared to our own human ability to understand speech. Some of the difficulties hampering progress in the fields of speech recognition and understanding stem from the many sources of variation that occur during human communication.

One of the sources of variation that occurs in human communication is the different ways that words can be pronounced. There are many causes of pronunciation variation, such as: the phonetic environment in which the word occurs, the dialect of the speaker, the speaker's age, the speaker's gender, and the speaking rate. Some researchers have shown improvements in speech recognition performance on a read-speech task through the use of explicit pronunciation modeling, while others have not shown any significant improvements.

This thesis presents an algorithm for the construction of models that attempt to capture the variation that occurs in the pronunciations of words in spontaneous (i.e., non-read) speech. A technique for developing alternate pronunciations of words and then estimating

the probabilities of the alternate pronunciations is presented. Additionally, we describe the development and implementation of a spoken-language understanding system called the Berkeley Restaurant Project (BeRP). Multiple pronunciation word models constructed using the algorithm proposed in this thesis are evaluated within the context of the BeRP system. The results of this evaluation show that the explicit modeling of variation in the pronunciation of words improves the performance of both the speech recognition and the speech understanding components of the BeRP system.

Contents

List of Figures

List of Tables

Acknowledgements

This work would not have been possible without the support and expertise of my friends and colleagues at the International Computer Science Institute including: Krste Asanovic, Jim Beck, Jeff Bilmes, Bryan Costales, Steve Greenberg, Dave Johnson, Brian Kingsbury, Phil Kohn, John Lazzaro, and Su-lin Wu. My research has especially benefited from frequent discussions with members of ICSI's speech group including: Dan Jurafsky, Ron Kay, Yochai Konig, Nikki Mirghafori, Nelson Morgan, Jonathan Segal, Andreas Stolcke, and Gary Tajchman. I will always be indebted to Nelson Morgan, my friend and research advisor for the past four and a half years, who was kind and patient enough to put up with my constant stream of questions and bad jokes, and who is responsible for the majority of my education in the field of speech recognition. I would also like to acknowledge Jerry Feldman, Director of ICSI, for providing an exceptional environment for learning and research.

I was also fortunate to have been part of several collaborations between ICSI and other researcher groups. From SRI, Mike Cohen gave valuable insights and recommendations on a draft of this thesis. Liz Shriberg, at U.C. Berkeley and SRI, provided guidance on the construction of the Wizard system. Hervé Bourlard of Lernout & Hauspie Speech Products has helped me in my attempt to understand some of the fundamental concepts relating to Neural Networks as probability estimators. Hynek Hermansky of Oregon Graduate Institute has helped me to grasp some of the concepts in the field of signal processing and has provided encouragement and all-around good advice during this research. I have also been fortunate to have been in close collaboration with Mike Hochberg, Steve Renals, and Tony Robinson at Cambridge University Engineering Department.

My education at U.C. Berkeley has been guided by Professor William S-Y. Wang. Bill Wang introduced me to computers and has been my advisor, my tennis partner, and my good friend for the past eight years.

The greatest debt I owe is to my family, especially my wife Wendy and my daughter Natasha, who expended as much effort as I did and sacrificed even more for the completion of this work.

This work was partially funded by ICSI and an SRI subcontract from ARPA contract MDA904-90-C-5253. Partial funding for the BeRP system development came from ESPRIT project 6487 (The WERNICKE project).

Chapter 1

Introduction

The goal of automatic speech recognition is to identify a word (or a sequence of words) in response to a person's voice. There are many applications in which a speech recognizer would be of great benefit, such as in automatic dictation, aids for the physically challenged, automatic information services over the telephone, etc. Often when we communicate with each other, the words are not as important as the ideas we are trying to express. So, even more beneficial than a speech recognition system would be a system that could *understand* what a person is saying.

One of the early speech recognition systems was developed in 1952 at AT&T Bell Laboratories (Davis *et al.* 1952). This system recognized the ten digits when spoken in isolation over a telephone line by a single individual with an accuracy of 97% to 99%. Current state-of-the-art speech recognition systems are capable of achieving accuracies of 90% to 95% on speech recognition tasks using 5000 words (Paul & Baker 1992)[1]. These systems are less constrained than the early AT&T recognizer in that they can recognize speech from more than just a single individual and they do not require speakers to pause between words. In the past few years, researchers have begun to develop speech understanding systems that perform moderately well within limited domains, such as automated airline travel information (DARPA 1992).

While it is undeniable that significant progress has been made in the field of speech recognition/understanding over the past 40 years, it is evident to anyone who has used current state-of-the-art systems that the technology will need to progress significantly before it can begin to approach our own human ability to understand speech. A few of the challenges facing current systems include recognizing speech in a noisy environment or in a crowded room, and recognizing speech that is spoken in a normal, conversational style. The difficulty in overcoming these challenges stems from the many sources of variation which

[1] Systems with 20,000 words are just now being developed.

cause errors in computer speech recognition and speech understanding systems. Some of these sources of variation are:

1. Variation in the environment – noise, room reverberation, etc.
2. Variation in the acoustic communication channel.
3. Variation in the acoustic signal due to a speaker's age, gender, physical stature and physical condition, etc.
4. Variation in the pronunciations of words due to dialectal or regional differences among speakers, or to the phonological context in which the words occur.
5. Variation in grammatical usage or style.

Eliminating or even partially controlling just one of these sources of variation is a very difficult task. In this thesis, we focus on the fourth type of variation mentioned above – variation in the pronunciations of words. While we do not propose a general solution to the problem of modeling variation in pronunciation, we outline an attempt to enable a speech understanding system to be less sensitive to this kind of variation.

1.1 Variation in Pronunciation

There are two ways that one can vary the pronunciation of a word: by varying the suprasegmental characteristics of the word or by varying the segmental characteristics of the word. Suprasegmental characteristics are those aspects of speech that are primarily controlled at the larynx or below (Wang 1972). Examples of such features include stress, tone, intonation, and duration. The phrase "segmental characteristics of a word" refers to the specific sequence of phonetic segments that comprise a word.

If either the segmental or the suprasegmental characteristics of a word vary significantly from the canonical or expected pronunciation, the word may adopt a new meaning. For example, changing the first phoneme in the sequence of phonemes comprising the word "bat" creates a word with a new meaning – "cat." Varying the suprasegmental features of a word can also change the meaning of the word. For example, by shifting the stress from the initial syllable of the word "permit" [pérmit] to the final syllable [permít], the word changes from a noun to a verb. It is these types of meaning-changing variation that must be modeled accurately by speech recognition/understanding systems.

Other types of suprasegmental variation in pronunciation, such as vowel duration in English, have no effect on the meaning of a word. Thus the word "cat" can be pronounced as either [k ae t][2] or [k aeː t][3]. Since duration is not generally meaningful for English, we

[2]See Table 2.3 for an explanation of these symbols.

[3]The ː symbol is used to indicate a lengthening of the sound that it follows.

must design our speech recognition and understanding systems so that they are robust to this type of variation.

Whether a particular type of variation in pronunciation is meaning-changing depends on the language. For example, in Hungarian, duration is meaningful, as illustrated by the words [u j] "finger" and [uː j] "new,"[4] and Estonian maintains a three-way duration contrast (Wang 1971):

kalas	kal˙as	kalːas
"in the fish"	"shore"	"he poured"

When presented with these Hungarian or Estonian words, a native speaker of American English may not be able to discriminate between them because duration is not distinctive in English.

Just as linguistically meaningful variation must be referenced to a particular language, so too must it be referenced to a particular speech recognition system. Currently, most speech recognition systems are trained on data from a specific task, such as dictated *Wall Street Journal* articles. Such a recognizer will perform poorly if it is used without modification to recognize speech obtained from a different task, such as one which uses non-read (i.e., spontaneous) speech. Other researchers (Cohen 1989) have pointed out the need to optimize models of phonological variation with respect to a particular speech recognition system. We cannot presume that all speech recognition systems "hear" speech in the same way. Some systems may use acoustic sub-word models that can capture much of the variation that occurs in vowels for example, and thus, such a system may not need to model this variation at the word level. For another system that may not be able to model the vowels as accurately at the sub-word level, the variation might need to be modeled at the word level. Therefore, if we wish to model pronunciation variation for the purposes of automatic speech recognition/understanding, we must ensure that we model these variations from the point of view of the particular recognition system being used.

Considering the wide range of variation that occurs in natural spontaneous speech (Butzberger *et al.* 1992), it seems obvious that word models that allow more than one pronunciation for a word should be better than models that allow only a single pronunciation per word. For example, a single-pronunciation model for the word "the" can only represent either the pronunciation [dh iy] or the pronunciation [dh ax], whereas a multiple-pronunciation model could represent both pronunciations.

Despite the seemingly obvious advantage of multiple-pronunciation word models, there has not been clear evidence that the use of such models can improve the performance of speech recognition systems. Some researchers (Lee 1989) have not shown any improvements in recognition performance through the use of explicit modeling of multiple pronunciations. Others (Cohen 1989) have demonstrated significant improvements in performance on large-vocabulary speaker-independent recognition systems.

[4]Thanks to Anita Liang for this example.

The construction of word models that attempt to capture the variation that occurs in the pronunciation of a word introduces many difficulties. For example, how does one derive alternate pronunciations for a word? And how do we represent the fact that certain pronunciations are more likely than others?

1.2 The Berkeley Restaurant Project

We have constructed a speech understanding system with which to explore the issues surrounding the modeling of variation in pronunciation and others topics in speech understanding and recognition. The Berkeley Restaurant Project (BeRP) is a medium-sized vocabulary, speaker-independent speech understanding system whose domain is knowledge about restaurants in the city of Berkeley. BeRP represents a coalescence of several research projects that have been underway at the International Computer Science Institute over the past five years. Research findings in the areas of robust acoustic feature extraction, connectionist speech recognition, pronunciation modeling, and natural language understanding have been incorporated into this system.

BeRP is a "mixed initiative" system, which means that either the user or the system may direct the interaction. The interaction begins with the system asking the user "How may I help you?" After the user records their response, the system will begin prompting the user in order to gain enough information to perform a database query. BeRP does not assume that the user's response will be related to its prompts; thus, the user is free to direct the interaction.

BeRP provides the structure needed to test not only the techniques proposed for the development of models of pronunciations, but ideas relating to natural language understanding, connectionist modeling, foreign accent detection and modeling, robust acoustical processing, and other research issues.

1.3 Hypothesis and Goals

The main question addressed in this thesis is whether or not explicit modeling of segmental and suprasegmental variation in the pronunciations of words within a spontaneous (i.e., non-read) speech understanding system will improve the performance of the system. Specifically, we propose a technique that will automatically derive (1) models of the durations of phonemes within particular phonetic contexts, and (2) models of the pronunciations of words as they occur in a corpus of training data. Our goal is to demonstrate that it is possible to model these kinds of segmental and suprasegmental variation in a speech understanding system (BeRP), thereby enhancing its performance.

The general algorithm developed in this work begins with a set of task-independent word models and through an iterative process, adapts them to produce a set of task-dependent word models. The adaptation process considers the specific characteristics of the speech recognizer and the pronunciations of words as they occur in a set of training data. There are

two motivating factors behind this algorithm. The first is to create a set of word models that better reflect how a particular speech recognition system "hears". The second motivating factor deals with the issue of portability. That is, we wanted a technique that would allow new speech recognition systems to be developed without the need for expert linguistic knowledge. Thus, the overall goal of this algorithm is to adapt the word models in a completely automatic, data-driven fashion.

We have implemented two versions of the multiple-pronunciation word model construction algorithm. The first implementation initializes the word model construction process with pronunciations obtained from a variety of sources, including pronunciation dictionaries and text-to-phoneme systems. The second implementation attempts to derive pronunciations automatically from a training corpus without the use of any linguistic constraints on the sequences of phonemes in the pronunciations.

Additionally, we present the details of the construction of the BeRP system and discuss several issues related to its implementation.

1.4 Outline of the Dissertation

Chapter 2 describes the pronunciation databases and acoustic databases used for the experiments presented in this dissertation.

Chapter 3 presents the details of the Berkeley Restaurant Project. First, we present an overview of BeRP and then some of the major components of the system. Finally, various implementational details are presented.

Chapter 4 presents a technique for the construction of explicit models of segmental and suprasegmental variation in pronunciation. The details regarding the incorporation of these models into the BeRP system are given next. Finally, we report on some preliminary experiments using a technique that attempts to construct models of the pronunciations of words automatically from training data.

Chapter 5 summarizes and presents some possible directions for future studies.

Chapter 2

Databases

This chapter contains information about the databases that were used for the experiments reported in this work. Two kinds of databases were used – pronunciation databases and acoustic databases.

2.1 Pronunciation Databases

As described in Section 4.2, the initial word models that are used for constructing multiple pronunciations are built from as many sources of pronunciations as possible. The multiple pronunciation word models used in the experiments reported in Section 4.5 were initialized from pronunciations obtained from the following five sources:

1. Lernout & Hauspie text-to-phoneme system
2. LIMSI-CNRS pronunciation lexicon
3. Resource Management
4. TIMIT
5. Handcrafted pronunciations (for some words)

Lernout & Hauspie text-to-phoneme system

The Lernout & Hauspie text-to-phoneme system produces phonetic transcriptions from the spellings of words. This is a very convenient method for obtaining pronunciations, especially for uncommon words or proper nouns such as restaurant names. The pronunciations produced from this system were used for the baseline single pronunciation experiments reported in Section 4.5.

LIMSI-CNRS pronunciation lexicon

The LIMSI lexicon was produced primarily for the 1992 *Wall Street Journal* continuous speech recognition task (Paul & Baker 1992) and contains pronunciations for approximately 10,000 words.

Resource Management

These pronunciations were developed by SRI and are distributed by NIST as part of the Resource Management speech recognition task (Price *et al.* 1988). They represent the most-likely pronunciations for the 1,000 words in Resource Management.

TIMIT

The TIMIT pronunciations were taken directly from phonetically hand-labeled speech data (see Section 2.2.1 below). There are about 6,100 unique words in the TIMIT database. Since many of the words occur more than once, there may be several alternate pronunciations for a word, and all of the pronunciations for each word were used.

Handcrafted Pronunciations

Approximately 500 words were transcribed by hand. This set consisted of words that could not be found in TIMIT, such as restaurant names and types of food.

The pronunciations from the five sources mentioned above were entered into a database of pronunciations. The entry for each word in the database contains all of the possible pronunciations for that word along with the source of the pronunciation. A sample entry from this database is given in Figure 2.1 (see Table 2.3 for an explanation of the symbols used in the pronunciations). The three-letter abbreviation that occurs at the beginning of each pronunciation indicates the source of the pronunciation (e.g. LHS – Lernout & Hauspie, LIM – LIMSI CNRS, HND – handcrafted).

```
LHS hh ae m bcl b axr gcl g axr
LIM hh ae m bcl b er gcl g axr
HND hh ae m bcl b er gcl g er
```

Figure 2.1: The pronunciation database entry for the word “hamburger.”

2.2 Acoustic Databases

2.2.1 TIMIT

The TIMIT database is a large database of speech that was collected at Texas Instruments and labeled by MIT (thus TI-MIT). The database was collected for the purpose of training speaker-independent phonetic recognition systems.

Speakers

TIMIT contains recordings from 630 American English speakers. Each speaker is classified as belonging to one of eight dialect regions (see Table 2.1). 70% of the speakers are male. Table 2.2 shows the distribution of the speakers according to ethnic background.

Dialect Region	Number of Speakers	Percent of Total
New England	49	7.7
Northern	102	16.2
North Midland	102	16.2
South Midland	100	15.9
Southern	98	15.6
New York City	46	7.3
Western	100	15.9
Army Brat (moved around)	33	5.2

Table 2.1: Geographical Distribution of Speakers in TIMIT

Ethnicity	Number of Speakers	Percent of Total
Native American	2	.03
Hispanic American	3	.05
Asian American	3	.05
Unknown	17	2.7
African American	26	4.1
Euro-American	578	91.7

Table 2.2: Distribution of Speakers in TIMIT according to ethnic background

Recorded Material

Each speaker in the database recorded 10 sentences. There are three types of sentences:

- 2 “sa” sentences. These two sentences were designed by SRI to elicit dialectal variations through the use of phonetic contexts in which such variations are know to

occur. The two sentences are “She had your dark suit in greasy wash water all year” and “Don’t ask me to carry an oily rag like that.” These sentences were spoken by all of the speakers in the database.

- 5 “sx” sentences. These “phonetically compact” sentences were designed by MIT to give a complete coverage of as many phonetic pairs as possible. MIT designed a total of 450 of these “phonetically compact” sentences and each speaker recorded five of the 450. Each sentence was thus recorded by seven different speakers.

- 3 “si” sentences. Since the small set of phonetically-compact sentences could not cover all possible phonetic contexts, a set of 1,890 sentences was selected by TI from the Brown corpus[1] (Kuchera & Francis 1967). Each speaker recorded three of these sentences; thus, all 1,890 sentences were spoken once.

Recording Environment and Processing

The speech was recorded digitally at a sampling rate of 20 kHz and then downsampled to 16 kHz. A Sennheiser close-talking microphone was used for all of the recordings.

The speech was initially labeled using an automatic procedure (Leung & Zue 1984) and then hand-corrected by linguists. The speech was labeled at both the phonetic and the word levels. Table 2.3 presents a list of the phones used to label the TIMIT database along with their IPA equivalents and an example word containing that phone. Most of the labels are phonemic. The rest were intended for labeling special acoustic events or acoustically-distinct allophones. Some of the labels need explanation:

stops The stops are labeled as a sequence of two events: a closure and a release. The closure refers to the period of time in which the articulators are closed preventing any air from escaping through the mouth. The release refers to the release of this closure. This was done in order to preserve boundary markings between these two acoustically-distinct events.

schwa Reduced vowels have four different allophones: back schwa [ə], front schwa [ɨ], retroflexed (or r-colored) schwa [ɚ], and voiceless schwa [ə̥].

/u/ In TIMIT the fronted version of /u/ is represented with a separate label – /ü/. The difference was determined by the position of the second formant. The back /u/ was used when F_2 was closer to F_1 than to F_3.

silence The silence that occurs at the beginning and end of each utterance is represented by the symbol “h#”. The symbol “pau” is used to mark pauses within a sentence. The symbol “epi” is used to label “epenthetic silence,” which marks “acoustically distinct regions of weak energy separating sounds that involve a change in voicing” (Seneff

[1]The Brown corpus is a one-million-word corpus assembled at Brown University in 1963-64. It contains samples of text from a wide variety of genres.

& Zue 1988), such as the gap that may occur between the /s/ and a semivowel or nasal as in the word “small” or “swift” or “prince.”

Phones in the TIMIT Database					
TIMIT	IPA	Example	TIMIT	IPA	Example
pcl	p^o	(p closure)	bcl	b^o	(b closure)
tcl	t^o	(t closure)	dcl	d^o	(d closure)
kcl	k^o	(k closure)	gcl	g^o	(g closure)
p	p	**p**ea	b	b	**b**ee
t	t	**t**ea	d	d	**d**ay
k	k	**k**ey	g	g	**g**ay
q	ʔ	ba**t**	dx	ɾ	dir**t**y
ch	t͡ʃ	**ch**oke	jh	d͡z	**j**oke
f	f	**f**ish	v	v	**v**ote
th	θ	**th**in	dh	ð	**th**en
s	s	**s**ound	z	z	**z**oo
sh	ʃ	**sh**out	zh	ʒ	a**z**ure
m	m	**m**oon	n	n	**n**oon
em	m̩	botto**m**	en	n̩	butto**n**
ng	ŋ	si**ng**	eng	ŋ̩	Wash**ing**ton
nx	ɾ̃	wi**nn**er	el	l̩	bott**le**
l	l	**l**ike	r	r	**r**ight
w	w	**w**ire	y	y	**y**es
hh	h	**h**ay	hv	ɦ	a**h**ead
er	ɝ	b**ir**d	axr	ɚ	butt**er**
iy	i	b**ee**t	ih	ɪ	b**i**t
ey	e	b**ai**t	eh	ɛ	b**e**t
ae	æ	b**a**t	aa	ɑ	f**a**ther
ao	ɔ	b**ou**ght	ah	ʌ	b**u**t
ow	o	b**oa**t	uh	ʊ	b**oo**k
uw	u	b**oo**t	ux	ü	t**oo**t
aw	$ɑ^w$	ab**ou**t	ay	$ɑ^y$	b**i**te
oy	$ɔ^y$	b**o**y	ax-h	ə̥	s**u**spect
ax	ə	**a**bout	ix	ɨ̥	deb**i**t
epi		(epen. sil.)	pau		(pause)
h#		(silence)			

Table 2.3: Phone Types Used

Availability

The TIMIT database is distributed by the Linguistic Data Consortium (LDC). The database is distributed on CD-Rom and contains speech waveforms, time-aligned phonetic and word transcriptions, and biographical information on all of the speakers.

To obtain the TIMIT database, contact Mark Liberman – Director LDC, 619 Williams Hall, University of Pennsylvania, Philadelphia, PA, 19104-6305, (215) 898-0141, email–myl@unagi.cis.upenn.edu.

2.2.2 The Berkeley Restaurant Project

The Berkeley Restaurant Project (BeRP) is a *knowledge consultant* whose domain is knowledge about restaurants in the city of Berkeley. Users interact with the system by asking questions via a microphone. The system recognizes the speech and then queries a database of restaurants and gives advice to the user based on such criteria as cost, type of food, and location of the restaurant. For more information on BeRP see Chapter 3.

Speakers

Recorded Material

WIZARD DATA: An initial set of 723 sentences was collected from 40 speakers during the spring of 1992. Table 2.4 show a breakdown of speakers according to native language and gender.

Native Language	Number of Speakers	Females	Males
American English	24	6	18
German	10	1	9
Italian	3	0	3
British English	2	0	2
East Indian English	1	0	1

Table 2.4: Speaker's Native Languages in the Wizard-collected Data.

The data was collected using a PNAMBIC[2] or Wizard of Oz methodology (Frazer & Gilbert 1991; Moore *et al.* 1991; Moore & Morris 1992). The Wizard methodology allows a system designer to collect data for a task without having the system "up and running." This involves getting the subject to believe that he/she is talking to a speech recognizer, when in fact there is a human operator in another room acting as the computer. This methodology is useful in order to ensure that the data collected will be consistent with how users will eventually interact with the "real" system.

The hidden operator or "wizard" was placed in another room and was listening and responding to the commands that the user was giving. The wizard used a mouse-based

[2]PNAMBIC = **P**ay **N**o **A**ttention to the **M**an **B**eh**I**nd the **C**urtain

interface to keep track of the user's requests and to interact with the user's screen (see Figure 2.2).

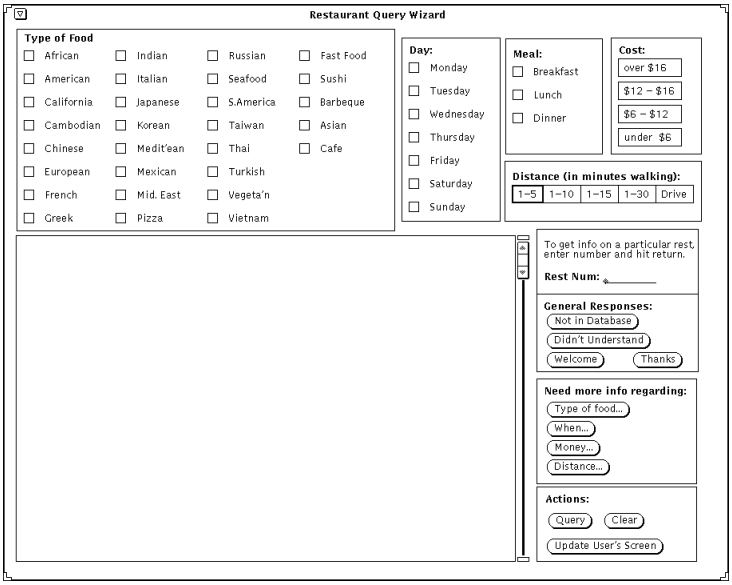


Figure 2.2: Wizard interface for initial data collection.

DC1 DATA: The Wizard-collected data was used to build an initial recognizer which was then substituted for the Wizard and used to collect additional data. The user interface for this data collection is shown in Figures 3.2 and 3.3. This additional data is referred to as DC1 (Data Collection session 1) data. During this first round of data collection using the actual recognizer, we collected 1,897 utterances from 44 new speakers. These speakers are listed according to their native languages and genders in Table 2.5.

DC2 DATA: A second round of data collection is currently underway. The recognizer for DC2 was built from the 42 American English speakers of Wiz and DC1. See Chapter 3 for more details.

Recording Environment and Processing

The data obtained during the Wizard data collection was recorded simultaneously over two microphones. The first microphone was a Sennheiser close-talking microphone and the second microphone was a Crown PZM table-top microphone which was used to collect data for future experiments on microphone robustness. The speech was recorded digitally at a sampling rate of 96 kHz. After recording, the speech was downsampled to 16 kHz.

All of the speech for DC1 was recorded digitally at a 16 kHz sampling rate using the Sennheiser close-talking microphone.

Native Language	Number of Speakers	Females	Males
American English	18	8	10
German	10	1	9
Italian	4	0	4
Mandarin	3	0	3
British English	3	0	3
Turkish	1	0	1
Japanese	1	1	0
Hebrew	1	0	1
Greek	1	0	1
French	1	1	0

Table 2.5: Speaker's Native Languages in DC1.

The recordings for all of the data collection sessions were made in a semi-quiet office with no attempt to suppress any environmental noise in the room. A signal-to-noise ratio of 42.25 dB was calculated (Hirsch 1993) for this data using the National Institute of Standards and Technology signal-to-noise ratio estimation software. This signal-to-noise ratio is sufficient for research purposes.

Chapter 3

The Berkeley Restaurant Project

3.1 Description

3.1.1 The Task

The Berkeley Restaurant Project (BeRP) is a medium-sized vocabulary, speaker-independent speech understanding system whose domain is knowledge about restaurants in the city of Berkeley. BeRP is similar to other spontaneous speech understanding systems that have been developed recently (Price 1990; Zue *et al.* 1990). Its primary purpose is to serve as a testbed for many ideas relating to speech recognition and understanding, including robust acoustic processing, connectionist modeling, foreign accent detection and modeling, automatic induction of multiple-pronunciation lexicons, and the tight coupling of advanced language models (such as stochastic context-free grammars) with the recognizer.

BeRP is a "mixed initiative" system, which means that either the user or the system may direct the interaction. The interaction begins with the system asking the user "How may I help you?" After the user records their response, the system will begin prompting the user in order to gain enough information to perform a database query. BeRP does not

currently assume that the user's response will be related to its prompts and thus the user is free to direct the interaction[1].

Although BeRP does not require a response that is relevant to its prompts, it monitors its prompts in order to maintain a context for interpreting user's responses such as "I don't care" or to resolve anaphoric references such as "it" in "it doesn't matter." Thus, if BeRP prompts the user with "What type of food would you like to eat?" and the user responds "it doesn't matter," BeRP will understand that the referent of "it" is "type of food" and will construct a database query that will search for all food types.

There were several requirements guiding the choice of the restaurant task:

1. The task should demonstrate the capabilities of our hybrid HMM/MLP continuous speech recognizer (see Section 3.3).

2. The task should support experiments in the areas of lexical modeling, natural language research, and accent modeling.

3. The task should not be too big. That is, the scope of the task had to be small enough that reasonable performance could be achieved at near-real-time speed.

4. The task could not be so trivial that it could be easily constructed with a simple speech recognizer such as an isolated word recognizer.

5. The task should be scalable in terms of vocabulary size and complexity. As the recognition system improves, the task should be able to grow in complexity so as to demonstrate the improvements.

The restaurant task proved to be well-suited to the requirements outlined above. Querying a database of restaurants is difficult enough that it cannot be handled with "off-the-shelf" speech recognizers, yet it is not too difficult to expect reasonable performance given the current state-of-the-art in speech understanding.

3.1.2 User Interaction

A hypothetical dialogue between a user and the BeRP system is shown in Figure 3.1. The results of this dialogue are shown in Figure 3.2. The dialogue begins with a prompt from the system in the form of a woman's voice. The user must then click on the "Record" button (see Figure 3.3) to speak back to the system.

[1] It would, however, make sense to bias the recognizer to favor replies which are related to the query, as discussed in Section 5.1.4.

Computer: Welcome to BeRP. How may I help you?

User: “I wanna have some Mexican food”

Computer: On what day would you like to eat?

User: “I’d like to go for lunch on Saturday”

Computer: How far from ICSI are you willing to travel?

User: “anywhere within walking distance”

Computer: How much money do you want to spend?

User: “doesn’t matter”

Computer: Here are some restaurants that meet your requirements:

Figure 3.1: A hypothetical dialogue between a user and BeRP.

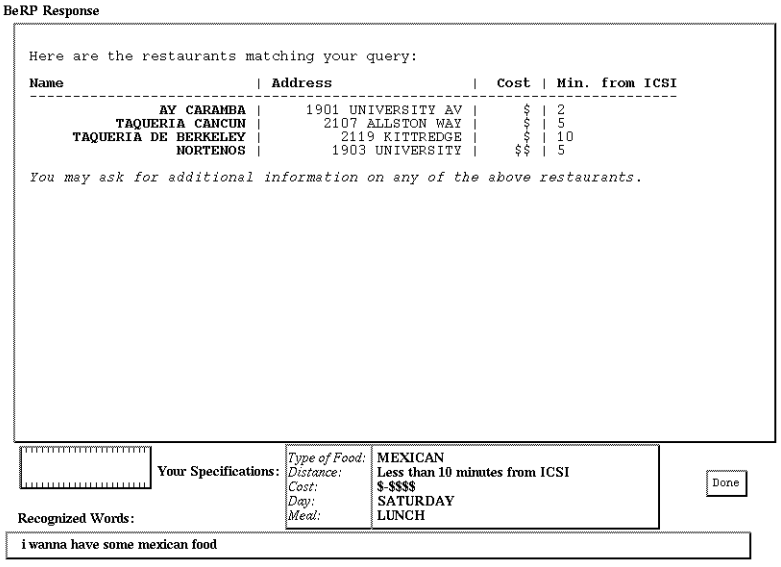


Figure 3.2: The results of a dialogue between a user and the BeRP system in which the user asks for information about Mexican restaurants serving lunch on a Saturday.

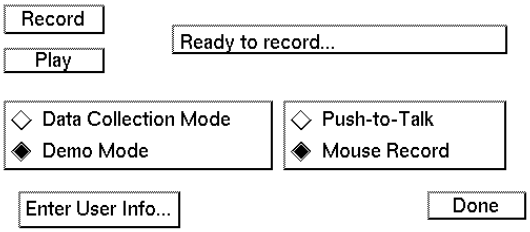


Figure 3.3: The user interface for the BeRP system. This interface is used when collecting data and when running the system in “demo” mode.

3.2 Acoustic Preprocessing

This section describes several of the many parametric representations of speech that are commonly used in speech recognizers today. We begin with a discussion of one of the most common parametric representation for speech recognition – Mel-cepstrum. Next we discuss a technique which, in some applications, has been shown (Chigier & Leung 1992) to give improved performance over Mel-cepstrum on clean speech - Perceptual Linear Prediction (PLP) (Hermansky 1990). The third parametric representation discussed in this section is a modification that can be applied to PLP (Hermansky *et al.* 1991) or Mel-cepstrum (Murveit *et al.* 1992) and is called RelAtive SpecTrAl (RASTA) processing. RASTA processing attempts to partially correct for the negative effects of convolutional noise, such as might be introduced by differences among communication channels.

One feature that all of these representations share is an attempt to incorporate knowledge from psychoacoustic research. For example, all three representations are based on a non-linear warping of the frequency spectrum. This warping is done on a bark (or Mel) scale in which the frequencies below 1 kHz are represented with greater resolution than those above 1 kHz.

3.2.1 Mel-Cepstrum

Mel-scaled cepstral coefficients are one of the most commonly used feature sets for modern speech recognizers. Cepstral coefficients are calculated from the inverse fourier transform of the short-term log spectrum. Mel-scaled cepstral coefficients are identical to normal cepstral coefficients except that the frequency axis has been warped to approximate the frequency scale of human hearing. That is, the frequency bands are spaced linearly below 1000 Hz and logarithmically above 1000 Hz (Zwicker 1961). This has been shown (Davis & Mermelstein 1980) to give improved speech recognition performance over cepstra calculated from linearly-spaced frequency bands.

3.2.2 PLP

One approach to dealing with the problem of recognizing speech from multiple speakers is to use an analysis technique which is effective at preserving linguistic information while suppressing speaker-dependent variations. The Perceptual Linear Prediction (PLP) analysis technique (Hermansky 1990) has been shown (Morgan *et al.* 1991a) to work well in this regard.

The PLP features are the cepstral coefficients of the autoregressive all-pole model of an auditory-like spectrum. The auditory-like spectrum used in PLP analysis is a bark-scaled critical band integrated power spectrum. Each band in the spectrum has been processed by an equal-loudness pre-emphasis and a cubic root nonlinearity to simulate the auditory intensity-loudness relation. The main difference between the mel cepstrum and the PLP cepstrum is in the method of spectral smoothing, which is achieved using cepstral truncation in the case of Mel-cepstrum and by autoregressive modeling in PLP.

3.2.3 RASTA

In Section 3.3.2 we discuss some of the initial difficulties one faces when building a speech understanding system. One of these difficulties lies in the fact that for a new task, there is often no data (or very little data) with which to train the speech recognizer, and data is very expensive to collect. We can partially make up for this lack of data by training a recognizer with speech from another task such as TIMIT (see Section 2.2.1). While databases of speech are readily available, there are possible drawbacks to using these sources of data. For example, it is likely that the acoustic conditions (room characteristics, microphone, etc.) under which the data was gathered will be different from the conditions of the new task. Such differences may appear to the recognizer as linear distortions of the acoustic signal which become an additive constant in the log spectrum. Many analysis techniques including Mel-cepstrum and PLP are known to perform poorly when presented with data that has such linear distortions.

We have integrated into BeRP an acoustic analysis technique called RelAtive SpecTrAl (RASTA) processing (Hermansky *et al.* 1992) which helps to minimize some of the effects of these types of linear distortions. The RASTA approach is conceptually simple and computationally efficient. The key idea is to suppress constant factors in each spectral component of the short-term auditory-like spectrum. This is done by replacing a conventional short-term absolute spectrum by a spectral estimate in which each frequency band is band-pass filtered by a filter with a sharp spectral zero at zero frequency. Since any constant or slowly-varying component in each frequency band is suppressed by this operation, the RASTA analysis is less sensitive to slow variations in the short-term spectrum.

We have primarily used RASTA processing on the logarithmic auditory-like spectrum of Perceptual Linear Predictive (PLP) analysis (see Section 3.2.2 above); however, other researchers (Murveit *et al.* 1992) have found that performing RASTA filtering on Mel-spectrum also works well. The constant factors that RASTA processing suppresses, represent convolutional "noise," i.e. the distortions introduced by the relatively time-invariant frequency response of the microphone and of the communication environment. The high-pass portion of the band-pass filter is expected to alleviate the effect of the convolutional noise introduced in the environment. The low-pass filtering is expected to help in smoothing out some of the fast frame-to-frame analysis artifacts.

It should be noted that the log-based RASTA-PLP is not robust to additive noise (Hermansky & Morgan 1992). However, there is work in progress on a modified version of RASTA-PLP that apparently does help with this kind of noise. Initial tests show that the revised form, which is based on a modified log-like nonlinearity, can suppress both convolutional and additive noise (Hermansky *et al.* 1993).

3.3 Recognizer

Although the speech signal is a continuously varying signal, it is commonly assumed to be quasi-stationary. The assumption is that during short intervals of time, such as 10

msec., the important features of the speech signal (i.e. the spectral estimates) will not vary significantly. Given this assumption, a 10 msec. interval of the speech signal can be represented by a single vector of acoustic features. The concatenation of all of the acoustic vectors for an utterance is used to represent the utterance for the purposes of training speech recognizers and for recognition.

The task of an automatic speech recognition system is to generate a string of words given a sequence of acoustic feature vectors. This task is traditionally represented as a maximization of the posterior probability of a model given a set of acoustic vectors, i.e. maximizing $P(M \mid X)$ (Bourlard & Morgan 1993). By Bayes Law, this probability can be expressed as:

$$P(M \mid X) = \frac{P(X \mid M)P(M)}{P(X)} \tag{3.1}$$

where $P(X \mid M)$ is referred to as the likelihood of the data given the model M, and $P(M)$ is the prior probability of M. As long as the parameters of the model M are fixed as is the case during recognition, the probability of the sequence of acoustic vectors – $P(X)$ is a constant and is typically ignored. The purpose of the recognizer is to compute the likelihood – $P(X \mid M)$. In Section 3.3.1, we will describe how we use a Multilayer Perceptron (MLP) to estimate the likelihoods needed in Equation 3.1. The likelihoods that are computed by the MLP are used by a Viterbi decoder to produce a recognized string of words.

3.3.1 Phonetic Likelihood Estimator

The current dominant approach to continuous speech recognition uses Hidden Markov Models (HMMs) to model sub-word speech units. Each model is represented as a finite state machine with probabilities assigned to the transitions between states and probability density functions for the observations associated with each state. These sub-word units can be either context-independent or context-dependent. The parameters for context-independent phone models (also called monophone models) are estimated from all instances of a phone in the training data regardless of the phonetic context in which the phone occurs. An advantage to using context-independent phone models is that there are usually a small number (40-60) of such models, and thus there is usually a sufficient amount of training data to estimate the parameters of these models. A disadvantage of context-independent models is that they are not constrained to model the coarticulatory effects that may be introduced by surrounding phones.

There are several types of context-dependent subword models. Each of these types attempts to model a phone within a particular phonetic context. The most specific context-dependent phone models are "word-dependent" models (Chow *et al.* 1986; Murveit & Weintraub 1988). A word-dependent phone model used in a particular word will have a different set of parameters from a model of the same phone in a different word. Another common context-dependent subword model is the triphone model (Bahl *et al.* 1980; Schwartz *et al.* 1984; Schwartz *et al.* 1985; Chow *et al.* 1986; Chow *et al.* 1987;

Kubala *et al.* 1988; Murveit & Weintraub 1988; Lee 1989) in which a phone is modeled within the context of the phones that immediately precede and follow it. The same phone in a different context will have a different model. Other possible context-dependent models include left-biphones and right-biphones in which only the context to the left or the right of the phone is considered when constructing the model.

The use of context-dependent subword models has led to significant reductions in error rates. The disadvantage of context-dependent subword models is that as the context becomes more specific, the number of models increases and it is often difficult to obtain enough training data to estimate the parameters of the models.

Whether a system uses context-independent or context-dependent subword HMMs, an HMM for an entire word can be constructed by concatenating a sequence of subword HMMs (see Figure 3.4). Likewise, an HMM for an entire utterance can be created by concatenating a sequence of word HMMs.

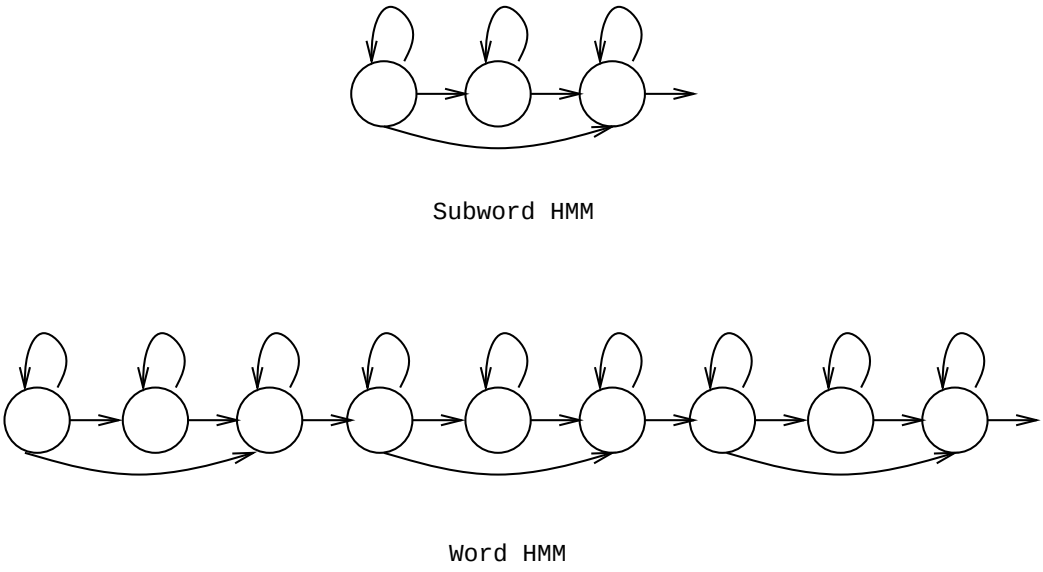


Figure 3.4: Examples of Hidden Markov Models for a subword and for a word.

As mentioned above, the purpose of the recognizer is to compute the likelihood – $P(X \mid M)$. If a model M is composed of a sequence of states q and the data X is composed of a sequence of acoustic vectors x, then to estimate $P(X \mid M)$ we need to compute $P(x \mid q)$ – the *likelihood* of a particular acoustic frame x given a particular state q. That is, given a vector x of acoustic features representing one frame of input, we need to compute the probability that this vector was generated by a given state q of an HMM. These likelihoods are given by the emission probabilities of the states of an HMM.

The Forward-Backward parameter estimation algorithm (Baum & Petrie 1966; Baum

1972; Baker 1975; Jelinek 1976; Levinson *et al.* 1983) is typically used for estimating the parameters that are responsible for generating the emission (and the transition) probabilities of the HMM. Recent work (Bourlard & Wellekens 1989; Bourlard & Morgan 1993) has shown how Multilayer Perceptrons (MLPs) can be used in conjunction with the Viterbi algorithm as a substitute for the Gaussian estimators that are normally used in the Forward-Backward algorithm for estimating emission probabilities. As shown in these reports, with a few assumptions, an MLP may be viewed as estimating the probability $P(q \mid x)$ where q is a subword model (or a state of a subword model) and x is the input acoustic speech data. Using Bayes' rule, this posterior probability may be represented as:

$$P(q \mid x) = \frac{P(x \mid q)P(q)}{P(x)} \tag{3.2}$$

This equation shows that the posterior probabilities that are generated by the MLP implicitly contain the prior probabilities of the states – $P(q)$. These priors may not match the priors that are implicit in the word models that are used during recognition (Bourlard & Morgan 1993) since the word models are typically constructed separately from the training of the MLP. By dividing both sides of Equation 3.2 by the prior probability of the state $P(q)$, we can eliminate the effects of the priors from the MLP, thus lessening the effect of the mismatch to the word models. Equation 3.3 shows the resulting scaled likelihood which can be used as the emission probability for a state q of an HMM.

$$\frac{P(q \mid x)}{P(q)} = \frac{P(x \mid q)}{P(x)} \tag{3.3}$$

Using an MLP to estimate the emission probabilities for the states of an HMM has several advantages over standard Maximum Likelihood Estimation (MLE) algorithms (Bourlard & Morgan 1993):

1. The MLP can be used to estimate likelihoods in a discriminative fashion (i.e. they are not only trained to accurately model the correct class, but to suppress incorrect classifications). Although other non-MLP training algorithms (e.g. Maximum Mutual Information) can be used to provide more discriminative training than MLE, they are more complicated and require more constraining assumptions.

2. The MLP requires no strong assumptions about the independence or the distributions of the acoustic features used for classification.

3. MLPs are highly parallel structures, which makes them particularly adaptable to special purpose hardware (Morgan *et al.* 1992).

Recognition experiments on the speaker-independent DARPA Resource Management database using context-independent classes (i.e., classes that are not conditioned on phonetic context) (Cohen *et al.* 1992), support the contention that these estimates lead to improved performance over standard estimation techniques, even when using the MLP in combination

with a fairly simple HMM. The MLP/HMM hybrid has yielded reasonable performance ($< 5\%$ word error with the standard perplexity 60 wordpair grammar).

BeRP MLP

The architecture of the MLP phonetic recognizer used in BeRP is the same as was used for our monophone Resource Management recognizer (Cohen *et al.* 1992). It consists of a simple feed-forward multilayer perceptron (see Figure 3.5) trained with the back-propagation training algorithm using a relative entropy criterion (Solla *et al.* 1988).

If the MLP accurately estimates posterior probabilities, then the outputs of the MLP will sum to 1. However, it very often happens that the network converges to a local minimum, in which case the outputs are no longer guaranteed to sum to 1. It is common for MLPs to use a sigmoidal output function:

$$g_k(x_n) = \frac{1}{1 + e^{-f_k(x_n)}} \tag{3.4}$$

where $f_k(x_n)$ is the value of output unit q_k (prior to the nonlinearity) for an input vector x_n. However, to ensure that the outputs sum to 1, the BeRP MLP uses a *softmax* (Bridle 1990) output function:

$$g_k(x_n) = \frac{\exp\left(f_k(x_n)\right)}{\sum_{i=1}^{K} \exp\left(f_i(x_n)\right)} \tag{3.5}$$

The input layer consists of 9 frames of input speech data. Each frame is composed of a vector of 8 RASTA-PLP coefficients (see Section 3.2.3) and their first derivatives and an energy coefficient and its first derivative. These 18 features were calculated over a 20 msec window of speech every 10 msec. The MLP used in the BeRP system also has 512 hidden units and 61 output units, one for each of the subwords (monophones) in the lexicon (see Table 2.3). Thus, the total number of parameters (weights) used in this recognizer is $(162 \cdot 512) + (512 \cdot 61) + 512 + 61 = 114,749$.

3.3.2 Task Adaptation

MLP Targets

Training an MLP using the error back-propagation training algorithm (Rumelhart *et al.* 1986)[2] requires that the training data be labeled. That is, each 10 msec frame of speech that is to be presented to the MLP must be labeled with one of the 61 phonemes from the lexicon. The fact that MLPs require this kind of labeling was initially thought to be a significant drawback to their use for speech recognition because of the prodigious amount of training data needed for training.

[2] An earlier form of this algorithm was discussed by Werbos (1974).

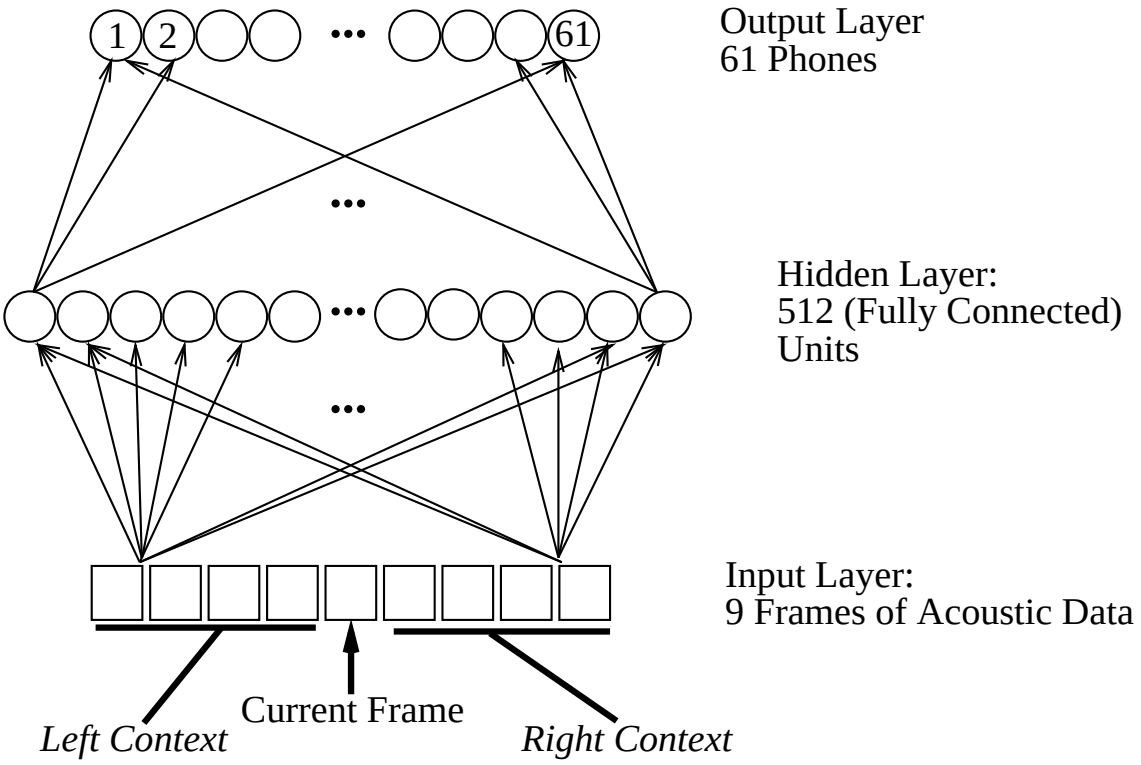


Figure 3.5: Phonetic Likelihood Estimator – MLP

While it is true that the training data must be labeled, we have found that it does not have to be labeled *by hand*. We can use a database of speech such as TIMIT (see Section 2.2.1) that has already been labeled as a starting point for an automatic labeling procedure.

The automatic labeling procedure begins by training an MLP on the TIMIT database. The training set used for this initial phone recognizer consisted of 8 sentences from each of the 630 speakers in the database for a total of 5,040 training utterances. During the training of the MLP the 5,040 utterances were divided into a training set of 4,540 utterances and a cross validation (Bourlard & Morgan 1993) set of 500 utterances. An MLP with 512 hidden units was trained using RASTA-PLP as described in Section 3.3.1.

Once the TIMIT MLP has been trained, an estimate of the phoneme probabilities for each frame of data is calculated by passing the training data through a forward pass of the MLP. These probabilities (likelihoods) are then used in a *forced* Viterbi alignment (Viterbi 1967; Bourlard & Wellekens 1990) which performs a probabilistic match between a sequence of states representing the pronunciations of the words in the training sentences and the estimates of the likelihoods of phonemes as given by the TIMIT MLP.

The result of the forced Viterbi alignment is a phone label for each frame of data in

the training database. Once the training data has been labeled, a new MLP can be trained. Successive iterations of forced Viterbi alignment followed by MLP training can be used to further refine the labels (see Figure 3.6). This embedded training procedure is discussed at length in Bourlard & Morgan (1993).

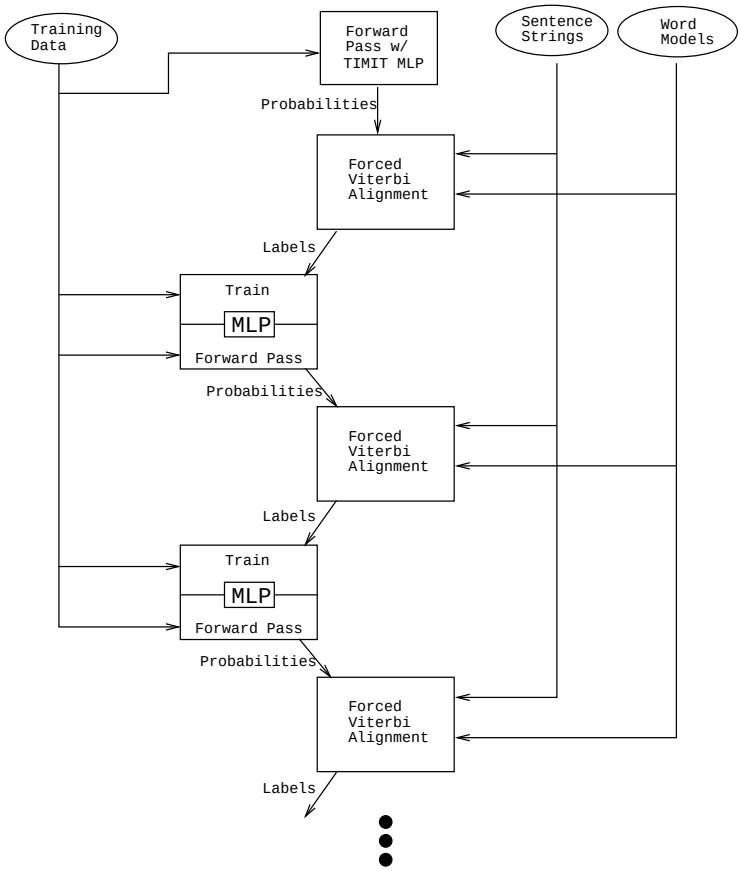


Figure 3.6: A schematic outline of the embedded training procedure for MLPs.

Insufficient Amount of Training Data

Chapter 2 describes how we began collecting data for the BeRP system using the "Wizard of Oz" methodology. Using the wizard system, we collected approximately 700 utterances. In order to automate the data collection process, we wanted to replace the wizard system with a preliminary form of the recognizer as quickly as possible. In order to train a recognizer that could replace the wizard system, however, we needed to have a fairly large amount of training data.

One solution to this dilemma is to train a recognizer on speech from a different task (for which there is ample data) and use that recognizer in the new task. Bourlard *et al.* (1993) showed that while a completely task-independent recognizer does not perform very well, the performance can be enhanced by retraining the task-independent recognizer with some speech from the new task added to its own data.

Our approach to the problem of constructing a recognizer for BeRP from insufficient training data was to initialize the weights of the BeRP MLP from the weights of the TIMIT MLP. When we train an MLP we typically initialize the weights from small random values. However, since 700 utterances is a relatively small amount of data[3] to use for training an MLP, initializing the weights from random values was not likely to result in a net that would provide good estimates of the phone probabilities needed for recognition. Thus, the weights for the BeRP MLP were initialized from the TIMIT MLP that was trained as described in the previous section.

Preliminary testing showed that a single pass through the set of 700 sentences was sufficient for training the BeRP MLP when the weights were initialized from the TIMIT MLP. We also tried a completely task-independent recognizer (i.e. just using the TIMIT MLP for recognition). We found that recognition using a completely task-independent recognizer gave worse performance than a recognizer trained with task-specific data but initialized with the TIMIT MLP weights. This result is consistent with the findings reported in other work on task independence (Bourlard *et al.* 1993). This new MLP was used in place of the wizard system for the second round of data collection as described in Section 2.2.2.

3.4 Natural Language Backend

Many problems are introduced when one moves from a speech recognition task to a speech understanding task. Rather than printing the string of recognized words, the speech understanding system must respond to the user in a way that is appropriate based on what the user said. In order for the system to respond appropriately, it must *understand* (in some sense of the word) what the user has said. Automatically deriving meaning from a string of words is a very difficult task even when the string of words is a grammatical sentence. It is even more difficult when the string of words is ill-formed due to errors by the recognizer. In fact, normal spontaneous speech is not grammatical (Butzberger *et al.* 1992), so even if the recognizer were perfect, the system must still be able to understand ungrammatical input.

An important component for any speech understanding system is the natural language backend. The natural language component is responsible for taking the output of the recognizer and transforming it into something useful. This section briefly describes work that has been performed by Dan Jurafsky of ICSI on the natural language backend for the BeRP system[4].

[3] It is more common today for large vocabulary speech recognizers to be trained on 4,000 to 40,000 utterances.

[4] For more details on this work please see (Jurafsky *et al.* 1993).

The BeRP backend accepts as input the word strings passed to it by the recognizer, and produces both database queries and appropriate responses to the user as output. Communication between the recognizer and the backend flows in only one direction. The recognizer passes a string of words to the backend, but no information is passed from the backend to the recognizer. Thus the recognizer and backend in the current BeRP system are *loosely coupled*.

This section briefly describes the five components of the BeRP backend:

- Grammar
- Parser
- Context Module
- Dialog Manager
- Database

Grammar

The BeRP grammar is a probabilistic context-free grammar augmented with simple semantic actions. The semantic rules are simple enough to make this a kind of context-free attribute grammar. The grammar currently contains approximately 1100 rules of the form

$$X \rightarrow \delta\{s\}[p] \tag{3.6}$$

where X is a non-terminal, δ is a (possibly empty) string of terminals and/or non-terminals, s is a semantic rule, and p is the probability associated with the rule. The probability is the conditional probability of the non-terminal X expanding to δ. The grammar is quite small, and currently only covers 70% of the training corpus sentences.

Parser

The BeRP backend uses both bottom-up and top-down chart parsers (Kay 1973) which use a simple dynamic programming algorithm to build a parse tree for each sentence that comes from the recognizer. The parse trees compute probabilities for parses and for prefixes, and build a semantic representation of each sentence on-line. They are on-line in the sense that each partial parse tree is augmented with semantics and a probability as it is built up. The top-down parser has the advantage that it is more efficient since it only accesses rules which will fit into the current parse tree for a sentence. The bottom-up parser has the advantage that it is more robust to ill-formed input from the recognizer. This robustness is due to the fact that the bottom-up parser accesses its rules based on the input sentence as opposed to top-down parsing in which rules are only accessed if they are consistent with the current left-to-right parse.

Context Module

The purpose of the Context Module is to augment the semantics produced by the Parser for each sentence, filling out all context-dependent and scope-dependent operators. The Context Module handles such phenomena as temporal deictics, negation, conversion of kilometers to miles, and pragmatically-dependent utterances such as “it doesn’t matter.” Deictic translations are required for such words as “now,” “today,” and “tomorrow.” Restaurant cost information is stored in four ranges represented as – "$", "$$", "$$$", and "$$$$", where each “$” represents six dollars. The Context Module is responsible for converting integral dollar amounts to these cost values.

Dialogue Manager

Figure 3.7 shows the architecture of the BeRP backend. The architecture is controlled by the *Dialog Manager*, which asks questions of the user in order to fill in a query-template which is used to query the database of restaurants. The current template has “slots” that are to be filled with the following information: cost per person, distance from ICSI, type of food, meal (breakfast, lunch, dinner or late-night), and day of the week.

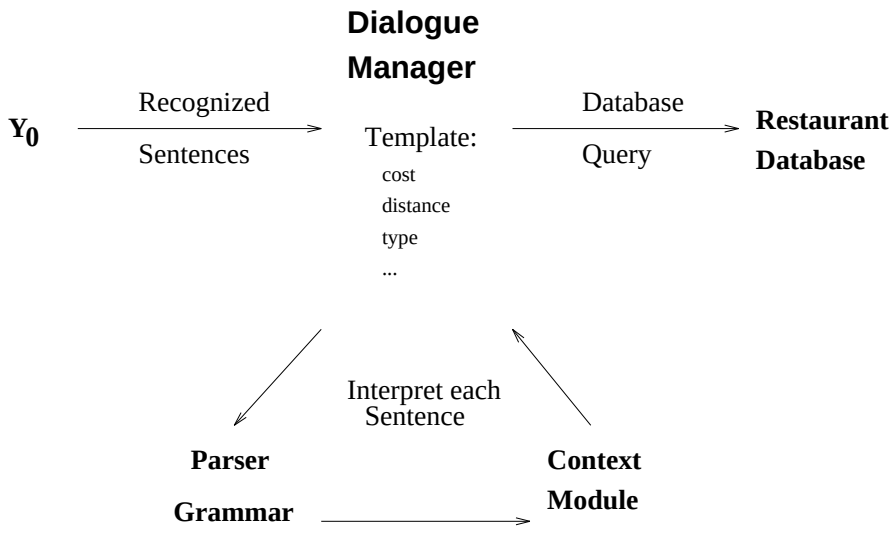


Figure 3.7: The Natural Language Component of BeRP

For each template slot the system prompts the user with a question. The recognized response to each question is passed from the recognizer to the *Parser*, which uses a bottom-up chart parser to produce a semantic interpretation for the sentence. After this interpretation

is augmented by the *Context Module*, the *Dialog Manager* uses it to fill the slots of the template. When the template is full, the dialog manager forms a database query and presents the query results to the user.

Restaurant Database

The database of restaurants is implemented using Postgres version 3.1[5]. It currently contains over 150 restaurants. For each restaurant the database contains the following information:

- Type of food served
- Other types of food served
- Average cost per person
- Business hours
- Distance from ICSI
- Name
- Address
- Nearest cross-street
- Phone number

Figure 3.8 shows a sample entry from the database.

Each restaurant record can be accessed by any of four keys: *cost*, *type of food*, *distance from ICSI (in minutes walking)*, and *business hours*. The information for each restaurant was obtained by telephoning each establishment. We used a number of sources for the list of restaurants, including Pitcher (1989).

Future Work

In the past, the fields of speech recognition and natural language processing have been kept fairly distinct. A number of researchers have called for more use of natural-language-backend information by the recognizer (Moore *et al.* 1989; Seneff *et al.* 1992; Goodine *et al.* 1991). Nonetheless, despite the trend in recent systems to tighten the coupling between the recognizer and the backend, the truly tightly-coupled approach – passing syntax-and-semantic-based word transition probabilities from the backend to the recognizer – is quite difficult and has not been successfully implemented.

The significance of the use of this high-level linguistic information during recognition is clear – it should significantly lower the language perplexity as seen by the recognizer, strongly improving recognition performance. High-level information is especially useful to people when in a noisy environment. The integration of information from syntax, semantics, and pragmatics with the acoustic/word level recognition may help to decrease the sensitivity that current systems exhibit to these types of environmental variation.

[5]Postgres is available via anonymous ftp from `postgres.berkeley.edu`.

```
Name            : CHEZ PANISSE
Type of Food    : CALIFORNIA
Other types     :
Address         : 1517 SHATTUCK AV
Cross Street    : CEDAR
Phone number    : 548-5525
Cost            : more than $16.00 per person
Min. walk from ICSI: 15
Hours:(B=breakfast,L=lunch,D=dinner)
Mon Tue Wed Thu Fri Sat Sun
--- --- --- --- --- --- ---
      D   D   D   D   D

Comments        : 'Gourmet' 'Reserve far in advance'
'Alice Waters masterpiece, Mediterranean and
Provencal influences'
```

Figure 3.8: A sample entry from the BeRP database.

The models of grammar used in most current speech understanding systems are quite simplistic. Since the standard recognition algorithms need language information in the form of word-transition probabilities, the most popular approach has been to use Markov chains, or *n*-gram models, which estimate the probabilities of n-word sequences from large corpora. Since an n-gram grammar cannot model non-local contexts, a probabilistic context-free or similar rule-based grammar will have a lower perplexity. A grammar which includes semantic information will have a still lower perplexity.

The grammar we are currently using in BeRP is a bigram grammar. This bigram grammar is derived from a semantic grammar in which semantic features are compiled into the rules of the grammar. We plan to eventually use extensive semantic knowledge, including the kind of semantic valence and constructional knowledge used in earlier work on on-line interpreters (Jurafsky 1992a; Jurafsky 1992b).

We are currently working on implementing a tightly-coupled system in BeRP. One of the most challenging design aspects of the tightly-coupled parser has been trying to achieve real-time performance. Because the recognizer requires word transition probabilities very frequently, the parser is called as often as several times every 10 milliseconds. Jurafsky has implemented a number of performance improvements for the parser, including performing parses in parallel, an efficient hashed rule-indexing mechanism, and designing a state-cache for efficient re-use of earlier parse states.

3.5 Implementation

Hardware

When using BeRP, one speaks into a microphone attached to a Gradient Technology Inc. DeskLab model 4014 analog-to-digital converter sampling at 16 kHz. The digitized samples are transferred via the SCSI port to a Sun workstation. The rest of the BeRP system runs on the workstation except for the phonetic likelihood estimator (the MLP) which, because of its computational requirements, runs on a special purpose parallel computer called the Ring Array Processor (RAP) (Morgan *et al.* 1992).

The workstation computes the acoustic features (RASTA) from the digital samples it receives from the DeskLab and then sends the RASTA features to the RAP over the ethernet. The RAP runs the MLP and sends the probabilities back to the workstation where the Viterbi decoding is performed. The recognized word string is sent to the BeRP backend (which also runs on the workstation) and the results are displayed to the user. This process is shown schematically in Figure 3.9.

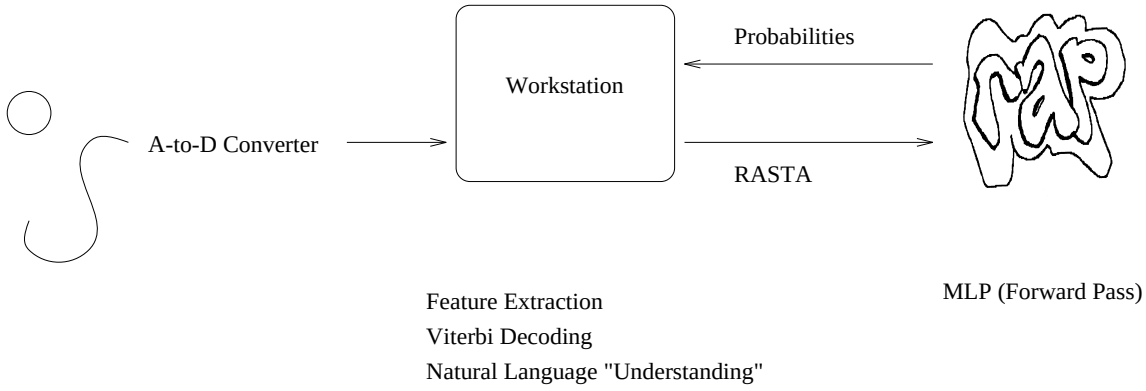


Figure 3.9: A schematic diagram of the implementation of the BeRP system.

Software

The BeRP system is implemented as a pipeline of programs in which the output from one program is piped into the input of the next program. There are four programs in the pipeline. The first program is responsible for the acoustic feature extraction, the second is responsible for running the MLP forward pass, the third does the Viterbi decoding, and the fourth is responsible for the natural language "understanding."

Currently, the acoustic features are sent to a program running on the SparcStation that is the host for the RAP machine. The program on the RAP host gathers the acoustic input vectors and sends them to the RAP via remote procedure calls. After the RAP has processed the input vectors, the resulting probabilities are piped to the Viterbi decoding software – Y_0 (pronounced "why naught").

Y_0 was initially developed at ICSI and since late 1992 has undergone major enhancements (and bug fixes) at Cambridge University Engineering Department as part of an ESPRIT-funded Basic Research project – the WERNICKE project (Robinson *et al.* 1993). Given the emission probabilities from the RAP, Y_0 runs the time synchronous Viterbi (i.e. dynamic programming algorithm), producing a recognized string of words. Y_0 consists of approximately 6500 lines of C++ code and includes features such as the ability to use multiple pronunciation word models, the ability to employ various pruning strategies, the ability to use a back-off bigram grammar, and the ability to do a forced Viterbi alignment.

Chapter 4

Lexical Modeling

One of the early techniques (Cohen & Mercer 1975) that was used to model the variation in the pronunciations of words used a "network" of allophones to represent alternate pronunciations. The allophone networks were constructed by applying a set of phonological rules to a dictionary of words. The phonological rules were constructed by expert linguists. The advantage to using allophone networks was that they could explicitly represent linguistic knowledge about the possible pronunciations of words.

There are several problems associated with the use of allophone networks. First, there was no mechanism to prevent the phonological rules from over-generating pronunciations. That is, the phonological rules may generate pronunciations that never occur. Another disadvantage is that there was no mechanism for explicitly representing the likelihood of the various alternate pronunciations. A major drawback to the use of allophone networks is that the development of the phonological rule sets used to produce the networks required a great deal of work by expert linguists.

Lee (1989) showed how probabilities could be associated with the different pronuncia-

tions represented in an allophone network. However, even with probabilities, Lee showed no improvements by explicitly modeling multiple pronunciations, and therefore he used only single pronunciation word models in his system.

Cohen (1989) pointed out that a possible reason why Lee (1989) could show no improvement through the use of multiple pronunciation word models may have been due to the fact that the models were too large. That is, the models were so large that the probabilities of the alternate pronunciations could not be accurately estimated given the amount of training data that was used. In his thesis, Cohen (1989) presents an approach to the construction of multiple pronunciation word models in which he attempts to maximize the coverage of the pronunciations on a set of training data while at the same time minimizing the overgeneration of pronunciations that rarely (or never) occur. Cohen was able to show a significant improvement in speech recognition performance using this approach.

The focus of this chapter is to outline a new algorithm for the creation of multiple pronunciation word models within the context of the hybrid HMM/MLP speech recognition system discussed in Chapter 3. One advantage to the approach presented here is that it is data driven, and thus it does not require an expert linguist to write a set of corpus-specific phonological rules in order to generate the initial, alternate pronunciations. Because it is data driven, it has the potential for being more portable to new speech recognition tasks.

The explanation of this new approach begins with a description of the techniques that we have been using to explicitly model phone duration and a possible way of improving this modeling. Then, Section 4.2 presents the algorithm that we use for creating multiple pronunciation word models.

4.1 Duration Modeling

Many researchers (Hochberg & Silverman 1993; Ferguson 1980; Russell & Moore 1985; Levinson 1986) have experimented with techniques for explicitly modeling the durations of phonemes for the purposes of automatic speech recognition. In all of these experiments it has been shown that explicit duration modeling improves speech recognition performance. In this section we describe the two approaches that we have used to model phoneme duration.

4.1.1 Context-independent Duration Models

In a Hidden Markov Model (HMM) speech recognition system such as the one we are using, phonemes are represented as a sequence of states linked together with transitions (see Figure 4.1) (Bakis 1976). A technique that has been used within the framework of neural network speech recognition systems (Bourlard *et al.* 1991; Bourlard & Morgan 1991) (see Section 3.3) to model phoneme duration is to create an N-state HMM for each phoneme where N is set equal to the minimum expected duration of that phoneme. If each state represents 10 msecs (which has worked well in our system [Morgan *et al.* 1991b]) then this model imposes a minimum duration of 30 msecs (3 states).

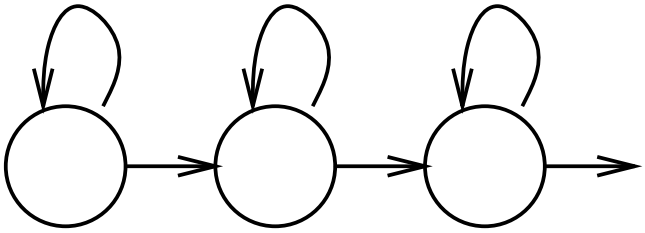


Figure 4.1: Simple duration model for a phoneme with a minimum duration of three states.

The minimum duration of a phoneme is approximated as one-half of the average duration of that phoneme. The average durations for the phonemes are calculated in terms of the average number of 10 msec frames assigned to that phoneme in the training corpus. When calculating the average duration for a phoneme, the phonetic context in which the phoneme occurs is not taken into account and thus these durations are "context independent." Appendix A presents the duration histograms for all of the phonemes used in our experiments. These histograms were calculated on the hand-labeled TIMIT database[1].

For our models, we assign self-loop and forward transition probabilities of 0.5 to the arcs exiting each state. Using a model with these transition probabilities and a minimum duration of m states, the probability of the duration d being equal to n states is given by Equation 4.1[2]:

$$P(d=n) = \begin{cases} 0.0 & \text{if } n < m \\ \begin{pmatrix} n-1 \\ m-1 \end{pmatrix} 0.5^n & \text{otherwise} \end{cases} \tag{4.1}$$

Figure 4.2 shows the duration distribution of Equation 4.1 for an HMM with a minimum duration of 5 states. The distribution shown in Figure 4.2 is very close to a Poisson distribution, which Hochberg & Silverman (1993) found to be a very good model of actual duration.

Figure 4.3 shows an overlay of the duration as predicted by our duration model and the actual duration distribution for the phoneme /aa/[3] from the TIMIT database. This graph shows that the predicted duration is very close to the actual duration.

[1] See Chapter 2 for more information on the TIMIT database.

[2] Thanks to Y. Konig for discussions on this point.

[3] See Table 2.3 for an explanation of this symbol.

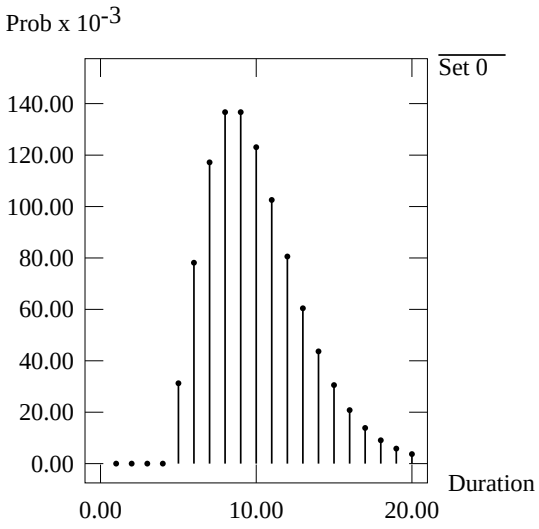


Figure 4.2: A graph showing the probability distribution function for the duration of a phoneme-HMM model with a minimum duration of 5 states.

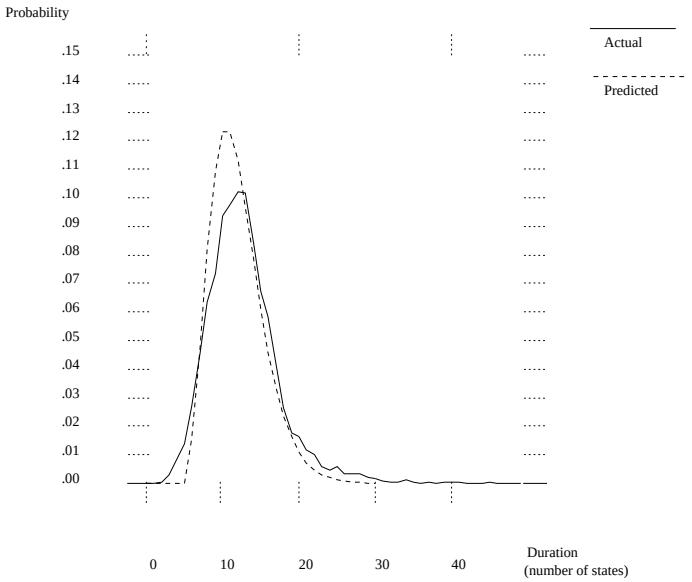


Figure 4.3: A graph comparing the actual duration distribution versus the predicted duration distribution for the phone /aa/ from the TIMIT database. This phone has an average duration of 12.4 states.

4.1.2 Context-dependent Duration Models

When calculating the duration models as outlined above, the duration measurements for each phoneme are calculated without regard to the phonetic context in which the phoneme occurred. This approach to modeling duration will tend to give poor estimates of the durations of many phonemes. For example, Table 4.1 presents data from Klatt (1975) showing the voice onset time (VOT[4]) for initial consonant clusters containing the phoneme /k/. This data illustrates how much variation can occur in the duration of a phoneme.

Cluster	VOT	Cluster	VOT
/k/	70	/sk/	30
/kr/	84	/skr/	35
/kl/	77		
/kw/	94	/skw/	39

Table 4.1: Data showing average duration of word initial consonant clusters containing the phoneme /k/. (All durations are reported in milliseconds. Standard deviation for /k/ is 11 msec.)

Context-independent duration models can also result in a poor match to actual durations when used to model the durations of phonemes in filled pauses. Frequently in spontaneous speech, talkers will fill potentially long intervals of silence with some speech sound, such as "uh". One of the characteristics of filled pauses is that the vowels that comprise the filled pause tend to have longer durations than those same vowels in a different word. For example measurements made on the BeRP database (see Section 2.2.2) show that the vowel in "uh", which is transcribed as [ah], had an average context independent duration of 104 msec. If the duration is calculated only from instances of [ah] taken from the filled pause "uh", then the average duration is 216 msec.

Context-dependent duration modeling attempts to capture the effects of various phonetic contexts on the duration of a phoneme. Possible contexts that may be used when calculating the duration of a phoneme include (in order from most specific to most general)[5]:

- Word specific – the duration for a phoneme is calculated only from samples of that phoneme in a particular word.
- Left and right phonetic context (triphone) – the duration for a phoneme is calculated only from samples of that phoneme when it occurs in the context of the specific phonemes to its left and right.

[4]Voice onset time is the time between the release of the stop closure and the onset of vocal cord vibration for the following sound.

[5]These categories of contexts are commonly used to model co-articulatory effects across phonemes (Lee 1989).

- Left phonetic context (left biphone) – the duration for a phoneme is calculated only from samples of that phoneme when it occurs in the context of the specific phoneme to its left.
- Right phonetic context (right biphone) – the duration for a phoneme is calculated only from samples of that phoneme when it occurs in the context of the specific phoneme to its right.
- No context (monophone) – the duration for a phoneme is calculated from all samples of that phoneme regardless of its phonetic context.

By restricting the calculation of durations to very specific contexts, we can more accurately model the duration of phonemes. However, given a fixed amount of data, the number of occurrences of a phoneme decreases as the context becomes more specific, making it more difficult to reliably estimate the durations in these contexts. Thus, there is a trade-off between more accurate models and the reliability of the estimates for those models.

Section 4.5 presents the results of experiments comparing the performance of context-independent durations versus context-dependent durations on a speaker-independent speech understanding task. When estimating the context-dependent durations, we used a "backoff" approach in which the most specific context (beginning with word-specific) was used whenever there were more than N occurrences of the phoneme in that context[6]. If there were fewer than N occurrences, then a less-specific context would be used until finally reaching a completely context-independent duration. In the experiments reported in Section 4.5, N was set to 10.

4.2 Pronunciation Modeling

The lexicon for a speech recognition system is composed of a set of models that represent the pronunciations of words. Most current speech recognition systems use lexicons comprised of a single pronunciation for each word that is to be recognized.

The construction of single-pronunciation word models is straightforward. One common technique that is used to construct these models is to use a text-to-phoneme system that can generate pronunciations based on the spelling of a word and a set of spelling-to-pronunciation rules. Using a text-to-phoneme system, large pronunciation dictionaries can be generated automatically. One advantage of this approach is that it is relatively easy to build pronunciations for new words.

When one considers the variety of realizations that a word may have depending on such factors as its phonological context, the dialect of the speaker, etc., it seems obvious that word models that allow more than one pronunciation for a word should perform much

[6]Perhaps a better approach to use in estimating the durations would be to smooth the estimates of the durations from all of the contexts using a technique such as *deleted interpolated estimation* (Jelinek & Mercer 1980; Lee 1989). However, a "backoff" technique such as the one we are using is a reasonable approach.

better in a speech recognition system than single-pronunciation word models. For example, a single-pronunciation model for the word "the" can only represent either the pronunciation [dh iy] or the pronunciation [dh ax][7], where a multiple-pronunciation model could represent both pronunciations.

Despite this seemingly obvious advantage, there has not been clear evidence that the use of multiple-pronunciation word models can improve the performance of speech recognition systems. Some researchers (Lee 1989) have not shown any improvements in recognition performance through the use of multiple-pronunciation word models. Others (Cohen 1989) have demonstrated significant improvements in performance on large-vocabulary speaker-independent recognition systems.

The construction of a model that attempts to capture the variation that occurs in the pronunciation of a word introduces many difficulties. For example, how does one derive alternate pronunciations for a word? Another difficulty arises when trying to represent the fact that certain pronunciations are more likely than others. Additionally, other researchers (Cohen 1989) have pointed out the need to optimize models of phonological variation with respect to a particular speech recognition system. The next sections present the approach we have developed in an attempt to overcome these difficulties.

4.2.1 System Independent Pronunciations

Although it is important to optimize models of phonological variation with respect to a particular speech recognizer, Cohen (1989) also points out that such optimization makes the models less interesting to those who would like to adapt them to a new system. The approach that he suggests is to construct an initial set of models based on system-independent data. Once the initial models have been constructed, they can then be adapted to a particular speech recognition system. Following Cohen's advice, the approach that we are developing begins with the construction of an initial set of HMMs representing alternate pronunciations for words, and then automatically adapts these HMMs to a particular speech recognizer. Finally, the adapted HMMs are combined with the phoneme duration models (either context-dependent or context-independent) to produce a multiple pronunciation lexicon.

The first step in the construction of a set of general pronunciation models for a lexicon is the accumulation of as many pronunciations as possible for each word that is to be modeled in the lexicon. We would like to accumulate as many pronunciations as possible in order to create an initial word model with the highest likelihood of capturing all of the phonological variation that may occur in the word. There are many sources of pronunciations that are available including: pronunciations derived from hand labeled speech, pronunciations from text-to-phoneme systems and pronunciations from dictionaries[8].

Once a database of pronunciations has been gathered, we can construct an initial pronunciation model for each word. The initial pronunciation model consists of a series of unique state sequences or paths: one for each of the alternate pronunciations of the word.

[7] See Table 2.3 for an explanation of these symbols.

[8] Section 2.1 presents the details of the pronunciation sources used in this work

The probabilities of each of the paths through the model are assigned uniformly, reflecting the fact that we have no information regarding the likelihood of these pronunciations. Given a model with N alternate pronunciations, the probability that is assigned to each of the arcs exiting from the initial null start state to the beginning state of each of the paths is $1/N$. Figure 4.4 shows an example of an initial word model for the word "and" with three alternate pronunciations.

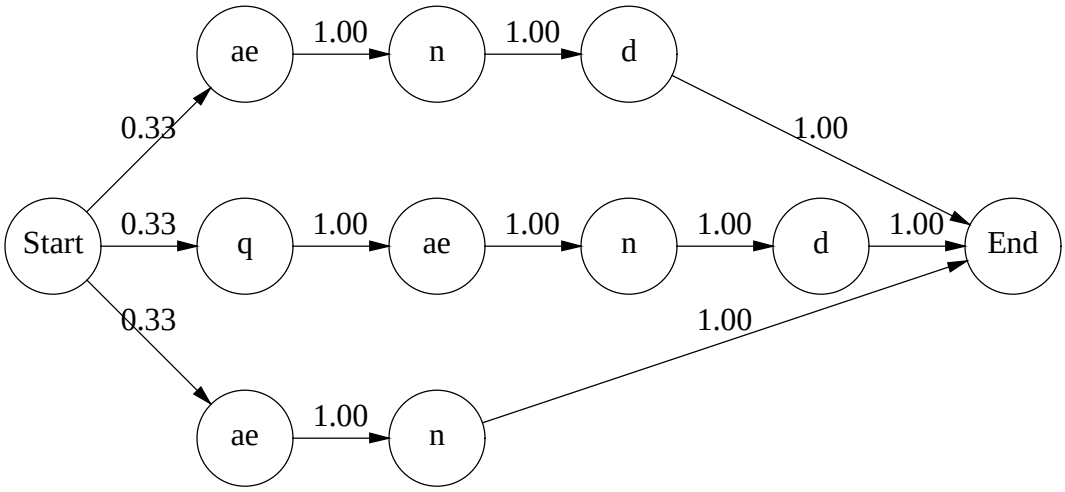


Figure 4.4: The initial form of a multiple-pronunciation word model for the word "and" with three possible pronunciations. (The symbol "q" represents a glottal stop.)

These initial word models are referred to as "system-independent" models. The system-independent models are used to initialize the construction of a set of system-dependent models as described in the next section.

4.2.2 Pronunciation Adaptation

Given a set of system-independent word models, the goal is to adapt these models to a particular speech recognizer, while at the same time replacing the *a priori* estimates of the likelihood of each of the alternate pronunciations of a word, with estimates that provide a better match between the speech recognizer and a set of training data. The two processes of adaptation and reestimation are carried out sequentially and may be iterated in order to further tailor the models to the speech recognizer and the training data. This algorithm is similar to the segmental K-means algorithm (Juang & Rabiner 1990; Pieraccini & Rosenberg 1989) that has been developed for estimating the parameters of Hidden Markov Models.

Adapting the pronunciation models

The adaptation procedure begins with a Viterbi (Viterbi 1967; G. D. Forney 1973; Bourlard & Wellekens 1990) alignment of the training data to the task-independent words models. During Viterbi alignment, a single path representing one of the alternate pronunciations of a word is chosen for each instance of the word in the training corpus. This path represents the pronunciation that best matched the outputs of the phonetic likelihood estimator (the MLP) for that particular occurrence of the word. Some of the pronunciations that are represented in the system-independent word model may never be chosen if they have a poor match to the outputs of the phonetic likelihood estimator compared to other pronunciations for the word.

Thus, the adaptation step produces a set of paths representing the pronunciations that had the best match between the outputs of the phonetic likelihood estimator and the alternate pronunciations of a word. These sets of paths can then be used to reestimate the likelihood of each of the alternate pronunciations of a word as described in the next section.

Reestimation of pronunciation probabilities

The technique that is used to reestimate the probabilities of each of the paths through an HMM is based on an algorithm for automatically inducing HMM structure from a set of samples (Stolcke & Omohundro 1993b). The algorithm begins with the construction of an initial HMM that just replicates the data (i.e. the paths representing the alternate pronunciations). Each path contains one state for each of the phonemes in the pronunciation. For example, Figure 4.5 shows the initial HMM for the following paths:

```
ae n
ae n
q ae n d
ae n d
ae n d
ae n d
```

In this HMM there are as many transitions out of the initial start state as there are unique paths to be merged. The probability of taking each of these paths is equal to the prior probability of the path. Thus, since three of the six paths contained the pronunciation – "ae n d", a probability of 0.5 is assigned to the transition from the initial start state to this path. This HMM model has the highest likelihood of producing the original set of paths.

Once the initial HMM has been constructed, it is made more general by successively merging states[9] (See Figure 4.6). The goal of the merging process is to induce a model from the data that is more general than the initial model (i.e., we want to "learn" from the

[9]In the version of the algorithm that we are using, we only consider merging states that have the same labels and we do not allow merges that would result in backward (i.e. right to left) arcs. However, in the full algorithm, there are no such constraints.

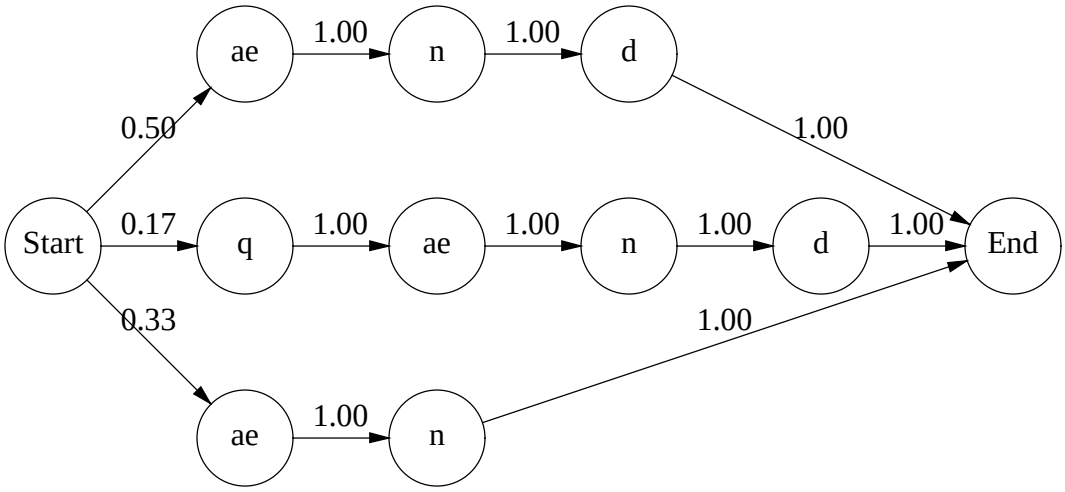


Figure 4.5: An unmerged HMM showing three possible pronunciations for the word "and."

data). This generalization is guided by a tradeoff between the likelihood of the model and a bias towards smaller models. By expressing the bias towards smaller models in terms of a prior probability, this tradeoff can be formalized using Bayes' rule:

$$P(M|x) = \frac{P(x|M)P(M)}{P(x)} \tag{4.2}$$

where $P(M|x)$ is the posterior probability that is to be maximized and $P(x|M)$ is the likelihood of the data given the model. $P(M)$ is the prior probability of the model. Since $P(x)$ (the probability of the data) is a constant for all of the models, we can ignore this term giving us:

$$P(M|x) \propto P(x|M)P(M) \tag{4.3}$$

As we begin merging states, the likelihood of the model, $P(x|M)$, will decrease. In order to maximize the posterior probability of the model, we need to offset any drop in the likelihood with a term that favors smaller models. Stolcke & Omohundro (1993b) found that by using a Dirichlet conjugate prior (Berger 1985) over the emission and transition probabilities of the model, they could produce an implicit bias towards smaller models.

The use of a Dirichlet prior corresponds to adding a number of "virtual" samples to the actual samples for the purposes of estimating the most-likely parameter settings. The same number of virtual samples is added to each of the possible emissions and transitions in the model creating distributions that are initially uniform. As the actual samples are added, the initially flat posterior distributions become more peaked in order to better fit the data.

The distributions will become more peaked around the maximum likelihood estimates of the parameters as more and more data is added. Given that the total amount of data is fixed, the fewer the number of states, the more data is available per state for the purposes of estimating the parameters, thus allowing the states to produce a better fit to the data. Stolcke & Omohundro (1993b) point out that "This phenomenon is similar, but not identical, to the Bayesian 'Occam factors' that prefer models with fewer parameters (MacKay 1992)."

Since the merging process reduces the likelihood of the model, but increases the prior probability, the merging can continue as long as there is an increase in the posterior probability.

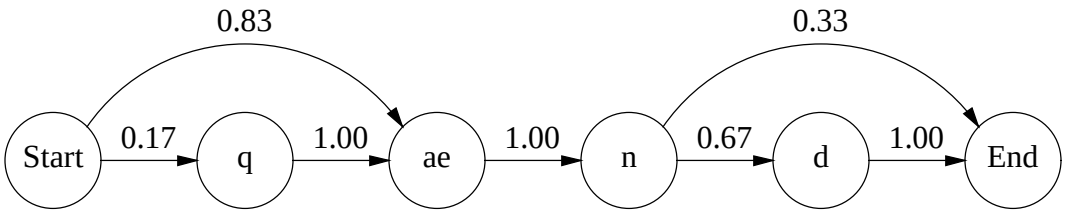


Figure 4.6: A merged HMM for the word "and."

As mentioned earlier, the goal of the HMM merging algorithm is to induce an HMM that is more general than the initial HMM that was constructed from the data. Thus, through the merging process, we hope to induce a model for a word that can generate previously unobserved pronunciations. For example, consider the following observed pronunciations for the word "have":

```
[hv ae v]
[hh ae v]
[hv ae f]
```

In this set of data, there are two phones that can occur at the beginning of the word – {hv,hh}, corresponding to a voiced /h/ and an unvoiced /h/ respectively. There are also two phones that can occur at the end of the word – {v,f}. There is one possible pronunciation – [hh ae f] that we may not have actually been observed in the data due to undersampling, yet we would want to allow it as a possibility. If the observed sequences are merged using the HMM merging algorithm, the unobserved pronunciation is induced automatically as show in Figure 4.7. The probability of this path through the model $(0.33 \cdot 0.33)$ is lower than the probability of any of the other paths through the HMM, which reflects the fact that we have not actually observed this path.

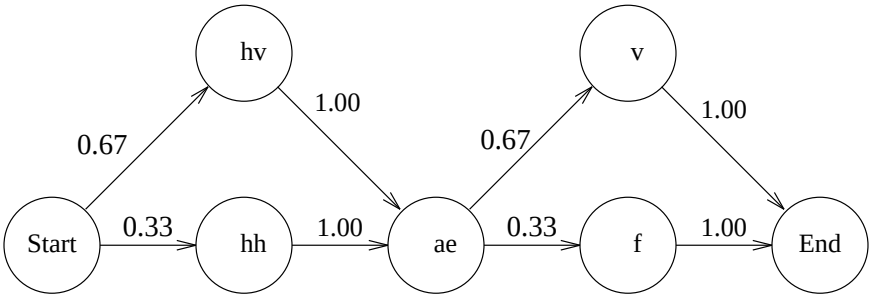


Figure 4.7: A possible HMM for the word "have" with a pronunciation – [hh ae v] that was not observed in the data.

4.2.3 Multiple Pronunciation Word Models

The final step in the process of producing a lexical model is to combine the merged HMMs with the duration models (either context-dependent or context-independent) to produce a multiple-pronunciation word model that can be used for recognition. The procedure that is used to combine the merged HMMs with the duration model is to replace each state (i.e. phoneme) in the merged HMM with N states, where N is the minimum number of states for the phoneme as given by the duration model.

It is slightly more complicated to combine the context-dependent duration models with the merged HMMs. The complication arises because a phoneme in a multiple-pronunciation word model may have several different phonemes to its left or right. For example, phoneme A may have a transition to phoneme B *and* to phoneme C. Given that there are two possible contexts to the right of phoneme A, the problem is how do we choose between B and C when determining the context-dependent duration to use for phoneme A? For the experiments reported here, we chose the shortest duration of all of the contexts in order to avoid creating a model that would be too long for some of the contexts. This approximation is fairly conservative and may be responsible for the fact that the context-dependent durations did not lead to improved performance for the multiple-pronunciation word models (see Section 4.5). Perhaps a better approach would have been to consider the probabilities of the contexts, and to use the duration for the context with the highest probability.

The self-loop and forward transition probabilities within the sequence of repeated states representing a phoneme are set to 0.5, except for the last state of the phoneme, which only has forward transitions whose probabilities are taken directly from the merged HMM. The fact that the last state has no self-loop is gives a slightly different HMM from that shown in Figure 4.1.

The new word models can now be used as input to another Viterbi alignment. The output of this second Viterbi alignment can be used as the input to HMM merging, and yet another set of word models can be constructed. This iterative process is shown schematically in

Figure 4.8, and coincides with the iterative process used for the generation of training labels for the Multi-layer Perceptron (MLP) as described in Section 3.3. Thus the generation of multiple pronunciation word models is easily integrated into the MLP's training procedure.

There is a similarity between this algorithm and the iterative Expectation Maximization (EM) algorithm (Baum *et al.* 1970; Dempster *et al.* 1977) that is used for approximating maximum-likelihood estimates from incomplete data. The Viterbi alignment step may be viewed as an approximation to the expectation step in which only a single label for each time frame is recorded. That is, while the Viterbi alignment uses probabilities internally when chosing the best path, only one label for each time frame in the sentence is output as a result of the alignment.

The two processes of HMM merging and retraining the MLP may be viewed as approximating the maximization step. That is, the MLP is maximizing the posterior probability of the phones given the acoustic data $P(Q|X)$ while the HMM merging is maximizing the probability of the word models given the pronunciations $P(M|X)$. Since each of these steps is independently performing a maximization we expect that each successive iteration will result in an overall reduction in error.

The previous sections have outlined a strategy for building probabilistic multiple pronunciation word models. This strategy will construct models that represent a locally optimal match, given the Viterbi assumptions and the error criterion, between the phonetic likelihood estimator (the MLP), an initial set of alternate pronunciations, and the training corpus. Section 4.5 presents the results of experiments in which this algorithm was used to generate multiple pronunciation word models in a speaker-independent spontaneous speech understanding system. The results show a significant improvement in recognition and understanding performance over single pronunciation word models on the same task.

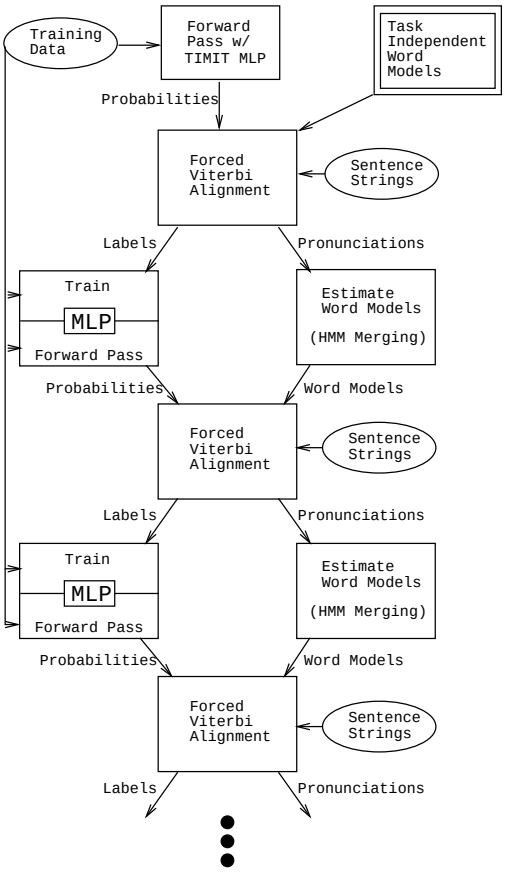


Figure 4.8: A schematic outline of the embedded training procedure for MLPs, modified to accommodate the construction of multiple pronunciation lexical models.

4.3 Incorporating Lexical Models into BeRP

This section presents the details of the process that we use to incorporate multiple pronunciation lexical models into the BeRP system. The algorithm we use (presented in Section 4.2) is a two-step process. The first step is to construct a set of "task independent" lexical models. These models are task independent because the probabilities associated with the various alternative pronunciations for each word have not been adapted to a specific task (i.e. all pronunciations are equally likely). The next step is to adapt the models to the BeRP task. This step is composed of an iterative procedure that is integrated with the embedded training algorithm outlined in Figure 3.6. The embedded training algorithm has been modified as shown in Figure 4.8 to accommodate the adaptation of the lexical models to a new task. The main difference between the new training procedure and the old procedure is the addition of a reestimation step after each forced Viterbi realignment. This reestimation step adjusts the probabilities of the pronunciations for each word as well as the topology of the word models.

4.3.1 Generating Task Independent Lexical Models

The first step in the process of constructing multiple pronunciation word models is to collect and process a set of pronunciations for each word.

Initial Set of Pronunciations

There are many sources of pronunciations that are available. Details regarding the sources of pronunciations used for BeRP are given in Section 2.1 and are listed again here:

1. Lernout & Hauspie text-to-phoneme system
2. LIMSI-CNRS pronunciation lexicon
3. Resource Management
4. TIMIT
5. Handcrafted pronunciations (for some words)

The goal in collecting the initial pronunciations is to cover as many possible alternative pronunciations for each word as possible. The larger the set of pronunciations, the more likely that they will cover the pronunciations found in the new task.

Mapping to the TIMIT phone set

One of the difficulties encountered when collecting pronunciations from several sources is that each source tends to have its own set of symbols for representing phonemes. In order to be able to use the pronunciations for all of the sources listed above, we had to construct a set

of mappings to transform the pronunciations as given by each source system into the phone set used for BeRP. These mappings were fairly straightforward, such as adding closures to stops for those systems in which closures were not represented separately.

The pronunciations from the L&H text-to-phoneme system, the LIMSI lexicon, and the Resource Management lexicon were mapped using a set of phonological rules designed for each system. The resulting multiple pronunciation lexicon contained all of the pronunciations from each source, and the pronunciations were all represented with a single set of symbols (the TIMIT symbols).

4.4 Loosening the Constraints

As discussed above, we use a Viterbi alignment to perform a probabilistic match between a sequence of states representing the pronunciations of words in a sentence and estimates of the likelihoods of phonemes as given by the MLP. During Viterbi alignment, a single path representing one of the alternate pronunciations of a word is chosen for each instance of the word in the training corpus. This path represents the pronunciation that best matched the outputs of the phonetic likelihood estimator for that particular occurrence of the word.

During the forced Viterbi alignment, the system is *forced* to choose the path through each word model that best matches the data. Therefore, the better the word models represent the range of possible alternate pronunciations that are found in the training data, the better the match will be during the Viterbi alignment. Thus, the goal in creating a set of multiple-pronunciation word models is to represent as many of the alternate pronunciations for each word as possible, resulting in a system in which the probabilities that are estimated by the MLP will find a better match to the pronunciations that are represented in the word models.

In the technique outlined above for creating multiple pronunciation word models, we began by collecting a database of pronunciations from many different sources. A significant amount of linguistic knowledge has gone into the construction of these sources. The sources typically consist of dictionaries of pronunciations or rule-based text-to-phoneme systems. One problem with these pronunciations is that they represent, for the most part, the "canonical" pronunciations of words with little consideration for how these words might be pronounced differently due to their phonetic context or because of the various deletion and assimilation processes that occur in natural spontaneous speech. For example, the word "don't" may have a dictionary pronunciation similar to [d ow n t], but in spontaneous speech the final /t/ can be dropped resulting in the pronunciation [d ow n].

Another problem with our initial set of pronunciations is that even with a large number of sources we still may not be guaranteed that these pronunciations will result in a good match to the phoneme probabilities that are emitted from the MLP. The reason for this is that the MLP has its own particular idiosyncrasies and doesn't perceive speech the same way that humans do. For example, the MLP may easily confuse the initial part of the diphthong /ay/ as in "bite" with the vowel /a/ as in "father." In this case, we would like to replace all occurrences of /ay/ in the pronunciations of words with the sequence /a/ /i/ so that our pronunciations are more consistent with the way the MLP "hears" speech. Thus,

if the pronunciations we have gathered from the various sources for the word "bite" have only the single diphthong /ay/ (and not /a/ /i/), the system will be forced to match /ay/ where it may have otherwise preferred match /a/ /i/. In this sense the system is constrained by the linguistic knowledge that has gone into the creation of the pronunciations. If we could loosen these constraints, we might be able to effect a better match between the MLP and the word models, resulting in an increase in recognition performance.

In order to test this hypothesis, we modified the algorithm used for the creation of multiple-pronunciation word models such that the pronunciations for the words are derived directly from the output of the MLP. This section describes some preliminary experiments we conducted in an attempt to verify this hypothesis.

4.4.1 Algorithm

The algorithm we use to create multiple-pronunciation word models that are free of the linguistic constraints imposed by our initial corpus of pronunciations is essentially the same as the algorithm presented above for the construction of multiple-pronunciation word models. The main difference between the two algorithms is in the initialization. The new algorithm begins with a phone recognition step using the TIMIT MLP. During phone recognition there are no word models; each phone is allowed to occur after any other phone in the lexicon. The only constraint that is placed on the generation of the phones is a duration constraint in which a phone must occur a minimum of N times where N is one half of the average duration of the phone as calculated from the TIMIT database. Since the phone sequence is free of the restrictions normally imposed by word models, we have called these phones "Loosephones."

The initial phone recognition step produces a Loosephone label for each frame of training data. These labels are used as targets for training a new MLP and for the generation of word models. The Loosephone word models are constructed using the HMM merging algorithm which was described in Section 4.2.2. In order to gather the Loosephone sequences that will be used in the HMM merging, we must know the beginning point and ending point for every occurrence of every word in the training database. For our initial experiments we obtained this word boundary information from the output of a Viterbi alignment that was run as part of the multiple-pronunciation word model construction process described above. After the Loosephones word models have been constructed and after training a net on the Loosephone labels, we can use the iterative training procedure just as it is described above.

4.5 Evaluation

4.5.1 Pronunciation Modeling

In this section, we present the results of several experiments with the BeRP system. These experiments were conducted to test both the effects of duration modeling and the multiple

pronunciation modeling technique. The MLPs for all of the systems tested were trained through two iterations of the embedded training procedure as described above. The initial labels were obtained from a forced Viterbi using the TIMIT MLP. There were 2,319 utterances in the training database. These were broken up into a set of 2,041 utterances for training and 278 utterances for cross-validation (see Section 3.3.2).

There are 364 utterances in the test set. These utterances were gathered from 8 speakers, 4 males and 4 females, each providing approximately 45 utterances. The speech is spontaneous and was not screened to remove any disfluencies or out-of-vocabulary words.

During Viterbi decoding (recognition) a simple "bigram" grammar is used to determine which words are allowed to follow each word and with what probability. This bigram grammar only allowed transitions for *observed* bigram pairs (i.e., the bigram was not smoothed) and thus is fairly constraining. One measure of the difficulty of a recognition task is given in terms of the test set "perplexity." Perplexity is roughly the average number of words that can follow any word in the vocabulary. (See Lee [1989] for a detailed discussion of perplexity.) Since there are out-of-vocabulary words and out-of-grammar word pairs (i.e. two-word sequences that are not in our grammar) in the test set, it is not possible to calculate its perplexity without making some assumptions about the probabilities of the out-of-vocabulary words and the out-of-grammar word pairs.

In order to get some idea of the perplexity of this test set, we calculated the perplexity on the subset of utterances in the test set that did not have out-of-vocabulary words or out-of-grammar word pairs. There were 227 utterances in this subset and the perplexity of these utterances is 10.6[10].

There were 48 out-of-vocabulary words in the test set (see Table 4.2). Most of these words occurred only once. The total number of out-of-vocabulary word tokens is 61, which represents 2.7% of the total number of words (2,241) in the test set.

apple	bagel	berp	block
buffets	can+t	caramba	caviar
chocolate	chowder	clam	cross
dessert	did	duck	durant
ethnic	fattening	frozen	healthy
keep	lo-cal	lobster	mid-priced
mykonos	necessarily	nineteen	omelet
parlors	pastries	peru	pie
ranges	rasa-sayang	rest	romantic
second	spot	sudanese	suggest
sundae	sweet-basil+s	takeout	taqueria
thing	tibetan	turkey	whoops

Table 4.2: Out-of-vocabulary words in the test set.

[10]Roughly one-third of the sentences in the test set had either out-of-vocabulary words or out-of-grammar word pairs. Thus, this task is much more difficult than this perplexity suggests.

Baseline Recognizer

The baseline system uses a single-pronunciation lexicon with context-independent durations. The pronunciations were built from the output of a commercial text-to-phoneme system[11].

Context-dependent Duration Models

Two lexicons were constructed to test the effectiveness of modeling phone duration in context as described in Section 4.1.2. When constructing word models with context-dependent durations, the duration of each phone was determined by an ordered table lookup beginning with the most specific context (word specific) and moving to less specific contexts (triphone, left-biphone, right-biphone, and finally, monophone), depending on the number of occurrences of the phone in that context. For these experiments, the minimum number of occurrences needed for any context was set to 10. If no context had more than 10 occurrences of a phone, then the context-independent duration was used.

Multiple Pronunciation Models

The multiple pronunciation models were constructed as described previously. Two lexicons were constructed, one with context-independent duration and one with context-dependent durations.

PRUNING WORD MODELS

For task-independent word models that have many different paths (i.e. pronunciations), after the merging process there will typically be only a few paths that are very likely. Many of the paths will have very low probabilities representing those pronunciations that rarely occurred in the BeRP data. We have found that we can improve the performance by pruning the unlikely paths from these bushy models.

The pruning algorithm that is used to eliminate unlikely paths is applied after the merging process has created a new word model. Once the new model is created, all of the unique samples (pronunciations) are processed by the model and are sorted according to the probability of the sample given the merged model. Beginning with the least-likely pronunciation (according to the model), we begin eliminating pronunciations until we have discarded enough pronunciations to account for some percentage of the probability mass. The percentage of the probability mass to discard is the parameter that must be determined experimentally. This pruning is shown graphically in Figure 4.9. In this graph, a sample pruning threshold of 0.1 is shown. All of the samples whose rank is to the right of the pruning threshold would be discarded. After pruning, the remaining samples are used to reestimate the parameters of the model.

[11]The text-to-phoneme system was generously provided by Hervé Bourlard of Lernout & Hauspie Speech Products.

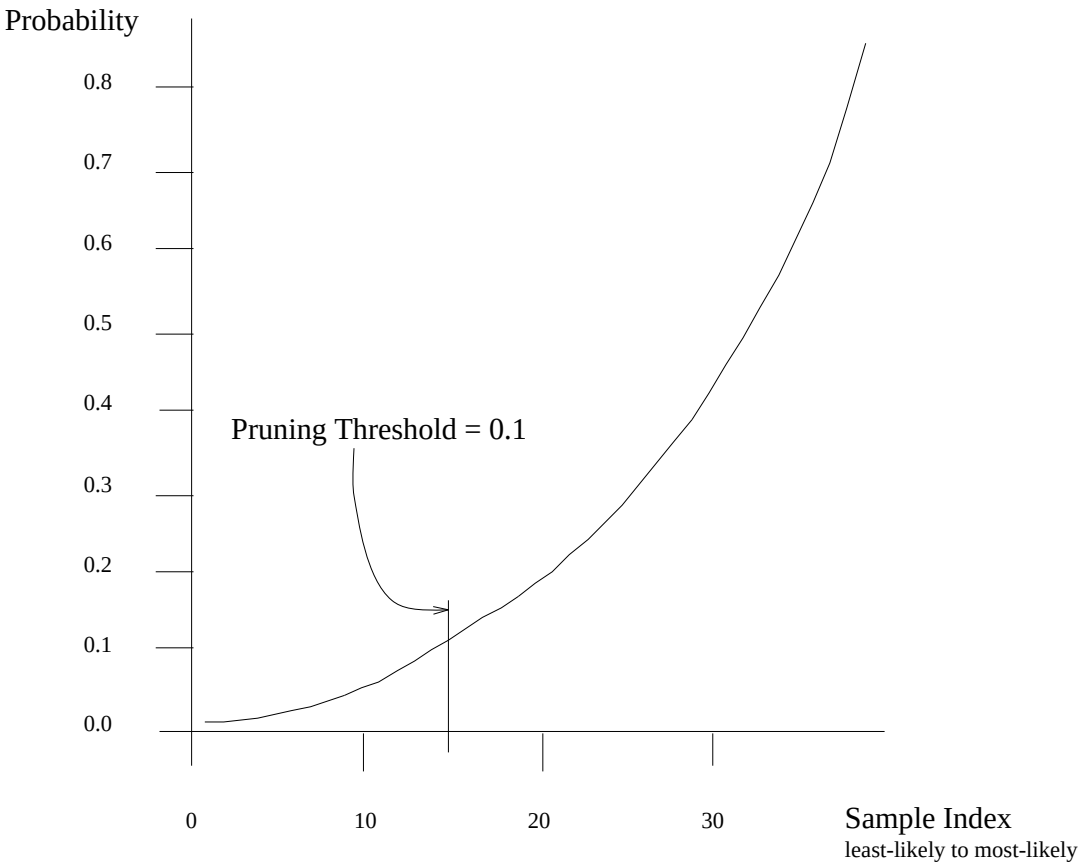


Figure 4.9: A probability histogram demonstrating probability mass pruning. The curve represents the *cumulative* probability of the pronunciations. The cumulative probability may not reach 1.0 because the model may be able to produce more pronunciations than were observed (due to HMM merging). The samples to the left of the pruning threshold would be eliminated.

Results

Table 4.3 shows the error rates for all four experiments. The percent error is calculated by taking into account all of the deleted, inserted, and substituted words as given in the following equation:

$$\%error = \frac{subs + dels + ins}{totalwords} \cdot 100.0 \tag{4.4}$$

Thus it is possible to have an error rate greater than 100.0%. While these error rates are much higher than for tasks involving read speech (as opposed to spontaneous speech), they are comparable to the initial results obtained at the Massachusetts Institute of Technology for their spontaneous speech understanding system – Voyager (Zue *et al.* 1990). These results are not as good as current state-of-the-art speech understanding systems for a couple of reasons. First, we are using an unsmoothed bigram grammar which is fairly constraining and second, we are using monophone sub-word models. Both of these could be improved, but this is outside the scope of this thesis.

In Table 4.3, the column labeled "Ins" shows the percentage of words that were inserted by the recognizer, "Dels" shows the percentage of words that were deleted, and "Subs" gives the percentage of words that were substituted. Note that for the context-dependent systems, it is only the durations that are context dependent, not the acoustic sub-word models.

System	Error Rate	Ins	Dels	Subs
Baseline	40.6	5.2	10.3	25.1
Base+CD Durs	38.6	4.6	10.6	23.4
Multi-Pron	32.1	7.4	5.7	19.0
Multi-Pron+CD Durs	32.1	7.3	5.9	18.9

Table 4.3: Results of BeRP experiments.

Table 4.4 shows a comparison between the four systems in terms of error rates only. According to scoring software provided by the National Institute of Standards and Technology which uses a Friedman two-way analysis of variance by ranks, there is no statistically-significant difference between the two baseline systems or between the two multiple-pronunciation systems. However, the differences are statistically significant when comparing each of the baseline systems to the corresponding multiple-pronunciation system.

	Baseline	Mult. Pron.
CI durations	40.6	32.1
CD durations	38.6	32.1

Table 4.4: BeRP performance with multiple-pronunciation word models and context-dependent duration models.

Table 4.5 presents a comparison between all four systems on all of the speakers in the test set. In this table, all of the speakers are ranked according to their performance in each of the four systems. Additionally, each of the systems is ranked according to its performance on all of the speakers. This table shows that one of the speakers (1G) seems to be much worse on average than all of the others. This speaker is a female speaker who had a tendency to use a significant number of filled pauses (um's and uh's) and a lot of restarts. For example, here is one of the utterances from this speaker:

> [uh] i don't need a ve-(getarian) no [loud breath] it doesn't have to be vegetarian [laughter] [uh] let's see [uh] never mind [laughter]

Utterances like this are very difficult to recognize because of the filled pauses, the non-speech sounds, the restarted phrases, and the truncated words.

\ SYSTEM					Av. pct
_____	mpi	mpd	cddr	base	
\					
SPKR \					Av Spkr R
	Percent	83.2%	76.7%	78.2%	76.0%	78.5%
19	Sys rnk	1.0	3.0	2.0	4.0	
	Spkr rnk	1.0	2.0	1.0	1.0	1.2
	Percent	76.0%	78.4%	73.9%	71.1%	74.8%
1J	Sys rnk	2.0	1.0	3.0	4.0	
	Spkr rnk	3.0	1.0	2.0	2.0	2.0
	Percent	78.7%	75.9%	69.6%	69.6%	73.4%
1B	Sys rnk	1.0	2.0	3.5	3.5	
	Spkr rnk	2.0	3.0	4.0	4.0	3.2
	Percent	71.3%	74.9%	70.9%	70.4%	71.9%
1Z	Sys rnk	2.0	1.0	3.0	4.0	
	Spkr rnk	4.0	4.0	3.0	3.0	3.5
	Percent	71.2%	67.6%	61.7%	54.1%	63.6%
1C	Sys rnk	1.0	2.0	3.0	4.0	
	Spkr rnk	5.0	6.0	5.0	6.0	5.5
	Percent	65.1%	67.8%	58.9%	56.6%	62.1%
1V	Sys rnk	2.0	1.0	3.0	4.0	
	Spkr rnk	6.0	5.0	6.0	5.0	5.5
	Percent	62.2%	61.0%	48.0%	48.6%	55.0%
1T	Sys rnk	1.0	2.0	4.0	3.0	
	Spkr rnk	7.0	7.0	7.0	7.0	7.0
	Percent	47.3%	50.1%	42.3%	39.9%	44.9%
1G	Sys rnk	2.0	1.0	3.0	4.0	
	Spkr rnk	8.0	8.0	8.0	8.0	8.0
Ave pcts		69.4%	69.1%	62.9%	60.8%	
Ave ranks		1.5	1.6	3.1	3.8	

Table 4.5: Ranking table for all systems and all speakers in the test set. The three numbers in each cell represent (from top to bottom): word accuracy, system's rank for the speaker, and speaker's rank for the system. (base = Baseline system, cddr = Baseline w/ Context-dep Durs, mpi = Mult Pron w/ Context-indep Durs, mpd = Mult Pron w/ Context-dep Durs)

Semantic Evaluation

Since BeRP is a speech *understanding* system, the word error rates do not provide a good description of the overall performance of the system. A better measure of how well the various systems performed on the test set is to calculate the number of utterances for which the system produced the correct "semantics," where "semantics" is defined in terms of database queries.

In Table 4.6 the semantic scores were calculated by comparing the semantics of each system to the "ideal" semantics for each utterance. The "ideal" semantics are the semantics (database queries) that would be produced if the natural language backend were perfect. These "ideal" semantics were constructed by hand by examining each of the test sentences and determining the correct database query for that sentence. The differences between the two context-independent duration systems and between the context-dependent duration systems in Table 4.6 are statistically significant[12] at the 0.05 level.

	Baseline	Mult. Pron.
CI durations	43.4	34.1
CD durations	44.2	36.3

Table 4.6: Semantic performance comparing to the "ideal" semantics. Scores are reported in terms of % incorrect.

The scores in Table 4.6 assume that the natural language backend is perfect and will produce the "ideal" semantics given a perfectly-recognized utterance. However, our current natural language backend is not perfect. We calculated the error rate for the backend by running the reference strings (the correct answers for each utterance in the test set) through the natural language component. The error rate for the backend on the reference strings is 18.1%. Approximately 18% of the utterances would not produce the "ideal" semantics even if the recognizer made no errors.

Given that the natural language backend cannot currently produce "ideal" semantics, we wanted to separate the performance of the recognizers from the performance of the backend. Thus, we compared the semantics of each system to the semantics as produced by the backend on the reference strings. These scores are presented in Table 4.7. The differences between the two context-independent duration systems and between the two context-dependent duration systems are statistically significant at the 0.01 level. Just as in Table 4.6, there is no significant difference between the baseline systems or between the multiple pronunciation systems.

Discussion

There are several interesting features in the experiments reported above. First, it is clear that the multiple-pronunciation word models perform much better than the single-pronunciation

[12] For the tests of significance in the semantic scores we used a normal approximation to a binary distribution.

	Baseline	Mult. Pron.
CI durations	39.3	27.7
CD durations	39.8	29.4

Table 4.7: Semantic performance comparing to the semantics produced by the natural language backend on the reference strings. Scores are reported in terms of % incorrect.

models on this task. The difference between the word-level scores on the baseline system and the multiple pronunciation context-independent durations system is 8.5% (40.6% - 32.1%), which represents a 20.9% reduction in the word-level error rate.

Another interesting feature is that while the context-dependent durations seemed to help the single-pronunciation word models, it made no difference for the multiple-pronunciation systems. One possible explanation for this may be that for the single-pronunciation models, the context-dependent durations provide a means for capturing some of the variation that is present in the pronunciations of the words. Once this variation is modeled explicitly, as in the multiple-pronunciation word models, there is no longer a benefit to detailed duration modeling. It may also be the case that the conservative approximation that was used when choosing a duration for a phoneme that had multiple contexts was a poor match for the data (see Section 4.2.3).

Inspection of the semantic scores shows that context-dependent duration modeling has no significant effect on performance[13]. This seems puzzling, especially given the word-level scores of the two single-pronunciation systems. A possible explanation for these seemingly contradictory conclusions is that while the context-dependent durations are helping to recognize more words correctly, they may not be words that are important semantically.

A preliminary examination of this hypothesis seems to support it. An examination of the ten most frequently misrecognized words in the two single pronunciation systems showed that all ten of the words were "function" words. In general, function words have little effect on the semantics given the natural language backend we are using. Table 4.8 shows these ten words and how often each was misrecognized. From the table, we see that there was a reduction in the number of misrecognitions for these words for the context-dependent duration system. This partly accounts for why the word level score is better for this system.

However, if we examine only the instances of restaurant names (which are very important semantically) we find that the number of misrecognitions did not change between the two systems (see Table 4.9).

This finding illustrates an important difference between a speech recognition system and a speech understanding system. If the goal is to recognize every word that a person says (such as might be the case in a dictation system), then it will be important to expend a significant amount of effort to accurately recognize all words, including the "function"

[13]While there seems to be a degradation in performance in the semantic scores for the context-dependent cases, there were only 3 differences between the two baseline systems in Table 4.6, and only 8 differences between the two multiple-pronunciation systems. The minimum number of differences needed for significance at the 0.05 level is 20.

	C.I.	C.D.
A	39	34
I	23	23
THE	18	16
ON	16	16
I+M	12	7
WHERE	11	10
HAVE	10	8
LET+S	10	11
GET	9	6
OF	9	8

Table 4.8: The top ten most frequently misrecognized words in the single pronunciation systems. C.I. is the context-independent system and C.D. is the context-dependent system.

	C.I.	C.D.
EDY+S	3	3
LA-TOUR-EIFFEL	3	3
RESTORAN-RASA-SAYANG	3	3

Table 4.9: The three most frequently misrecognized restaurant names in the single pronunciation systems. C.I. is the context-independent system and C.D. is the context-dependent system.

words. If the goal is to extract some kind of meaning from the words that were spoken (as in the BeRP system), then it is less important to concentrate on the "function" words than on those words that *matter* for the semantic interpretation of the utterance.

4.5.2 Loosephones

In our experiments with the Loosephones algorithm we ran five iterations of the embedded training/word model reestimation procedure. We used an MLP exactly like the one described in Section 3.3.1. Figure 4.10 shows the performance of the MLP on the cross-validation data after each iteration. These scores were very encouraging, especially when compared to the scores we get when using phoneme labels, which are typically around around 68% correct on the cross-validation data. However, there was very little change overall.

To get an idea of how much the labels are changing from one iteration to the next, we calculated the number of labels that are different at each iteration from the previous iteration. This graph is shown in Figure 4.11. The largest change (15.8%) occurs from the initial labels to the labels in the first iteration. By the fifth iteration, the labels are changing very little.

Figure 4.12 shows the word-level recognition scores after each iteration of the embedded

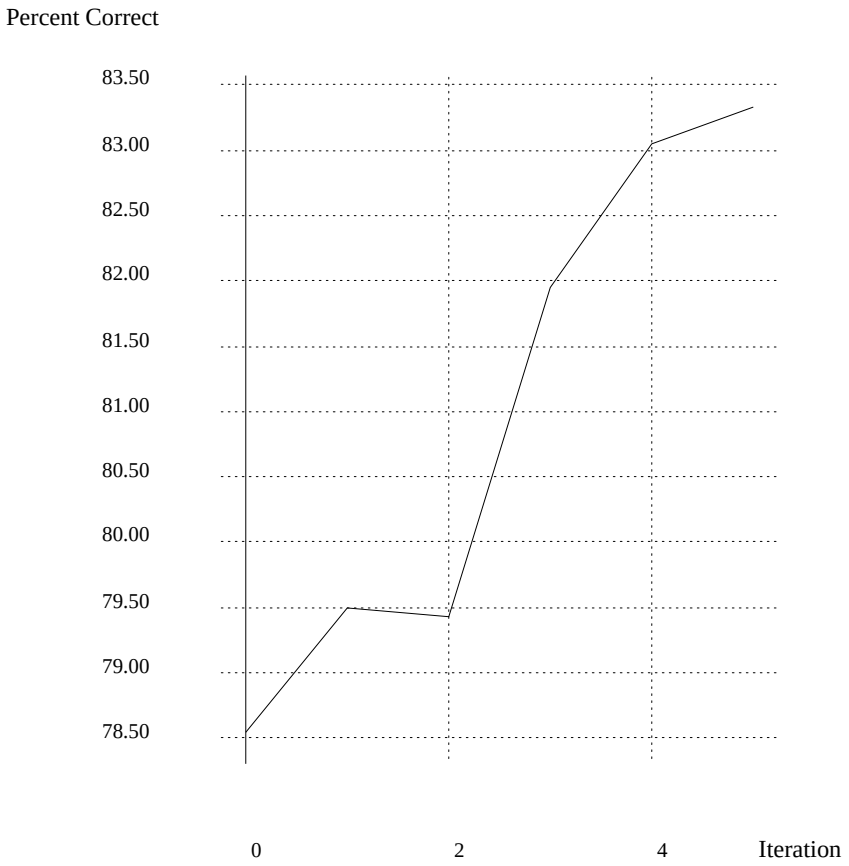


Figure 4.10: Frame level score on the cross-validation set after each iteration of embedded training using Loosephones.

training/word model reestimation procedure. These scores are for the same test set used in the experiments reported in Section 4.5 and are significantly worse than the scores we get when we initialize the algorithm from multiple sources of pronunciations as described in Section 4.2 above.

Modification

We hypothesized that a possible source of the errors in the Loosephones experiment described above was due to the size of the word models that result from this unconstrained algorithm. To test this hypothesis we continued the iterations using pruning during the HMM merging step as described in Section 4.5.1. Beginning with the iteration that performed the best at the word level (the fourth), we reestimated the word models using HMM merging with pruning (using a threshold 0f 0.05) and reran the Viterbi alignment without

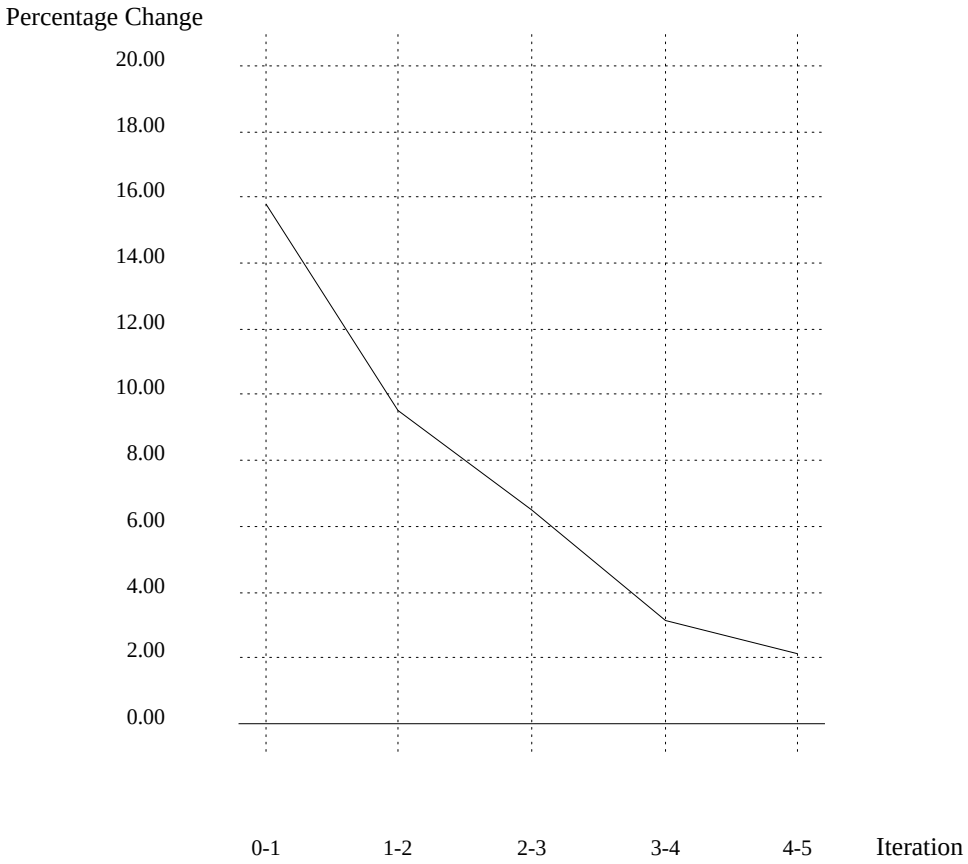


Figure 4.11: The percentage of labels that changed between successive iterations of the embedded training using Loosephones. The total number of labels in the training set is 683552.

retraining the MLP. We iterated this merging-pruning/Viterbi alignment procedure through two iterations. The word-level results are shown in Figure 4.13. It seems clear from these results that after two pruning steps the word-level scores are not significantly affected.

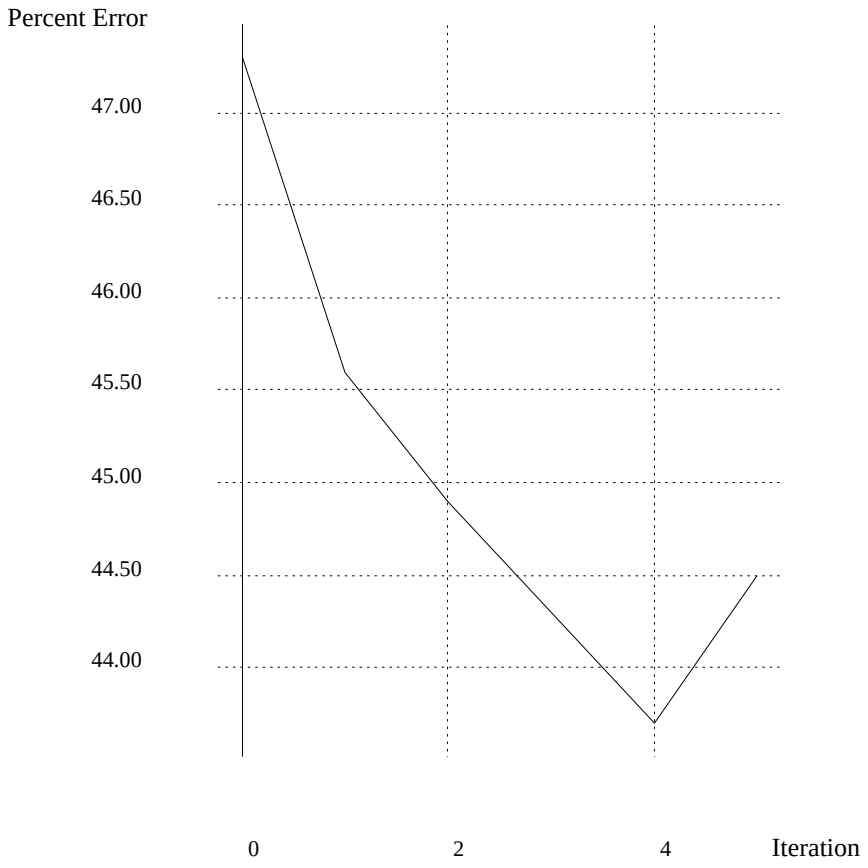


Figure 4.12: Word-level score on the test set after each iteration of embedded training using Loosephones.

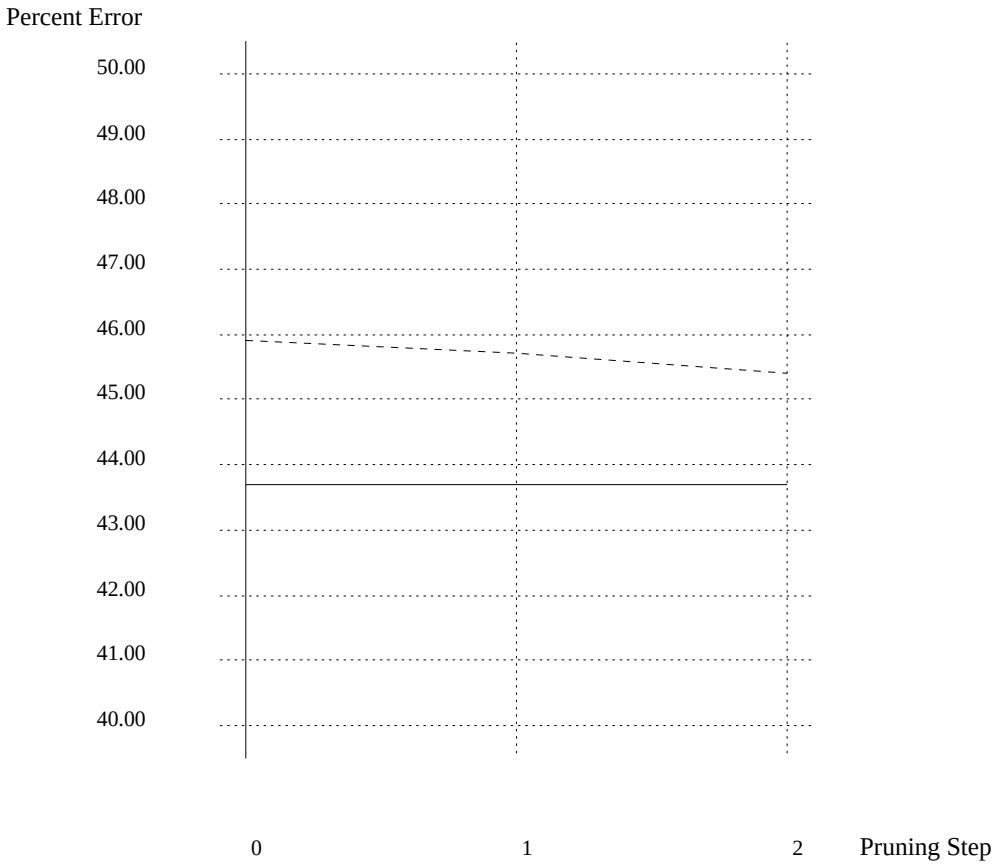


Figure 4.13: Word-level score after each pruning step starting at the fourth Loosephones iteration. Pruning step 0 is the same as the fourth iteration of the Loosephones, except that the models were pruned. The solid line represents the performance on the fourth Loosephones iteration.

Discussion

The results obtained in the Loosephones experiments do not support our original hypothesis that "relaxing" the constraints imposed by the initial Viterbi using pronunciations from the various linguistic sources would improve performance. In fact, these results would seem to indicate that making use of this type of linguistic knowledge is beneficial. It may be that because of the relatively small size of our training corpus, we could not reliably estimate the parameters of the large word models that are created in this algorithm. It is possible that given enough training data, this approach may yield improvements. It also may be possible that we need more than five iterations and that we have stopped the process too early, although this doesn't seem likely given the drop in word-level performance from the fourth to the fifth iteration.

A potentially useful application for this algorithm may be to use the Loosephones pronunciations for those words for which it is difficult to obtain pronunciations from other sources. This could be accomplished by running phone recognition on the word and using the sequence of phones that are recognized as the pronunciation for the word. It may be useful, in fact, to add the most-likely pronunciations for each of the Loosephones word models to the initial source of pronunciations as part of the standard multiple-pronunciation word model construction algorithm.

Another potential advantage to this approach is that it would allow for easy porting of recognizers to new tasks and new languages. Since the word models could be constructed completely automatically, there would be no need for labor intensive work by an expert linguist. In fact, the preliminary word-level scores for the Loosephones reported here were similar to the word-level scores from the baseline single-pronunciation system whose word models were produced from a text-to-phoneme system. Additionally, if we consider the semantic scores for the Loosephones, they do as well as the single-pronunciation baseline system as shown in Table 4.10.

If the Loosephones can be made to work better than the word models produced by a text-to-phoneme system, then they would have an advantage when constructing new speech recognition systems, especially when constructing systems for different languages. The results reported here for the Loosephones are preliminary, and we believe that with further experimentation they will improve.

	Baseline	Loosephones
"Ideal" Semantics	43.4	42.9
Backend Semantics	39.3	37.4

Table 4.10: Semantic performance for the baseline single-pronunciation system compared to the Loosephones system. The differences are not statistically significant. Scores are reported in terms of % incorrect.

Chapter 5

Summary and Conclusions

The main question that we have addressed in this thesis is whether or not explicit modeling of segmental and suprasegmental variation in the pronunciations of words within a spontaneous (i.e. non-read) speech understanding system would improve the performance of the system.

We have proposed a technique that will automatically derive (1) models of the durations of phonemes within particular phonetic contexts, and (2) models of the variations in the pronunciations of words as they occur in a corpus of training data. We have demonstrated that modeling these kinds of segmental and suprasegmental variation in a speech understanding system can enhance the system's performance.

Additionally, we have developed a speaker-independent spontaneous speech understanding system called the Berkeley Restaurant Project (BeRP). The primary purpose of the BeRP system is to serve as a testbed for many ideas relating to speech recognition and understanding, including robust acoustic processing, connectionist modeling, foreign accent detection and modeling, and the tight coupling of advanced language models (such as stochastic context-free grammars) with the recognizer. The BeRP system was used to test the techniques that we proposed for the modeling of pronunciation variation in spontaneous speech.

5.1 Summary

5.1.1 Duration Modeling

Our original hypothesis regarding the modeling of duration of a phone with the context of the surrounding phones was that such modeling would be able to capture more of the

variation in the pronunciations of words, thereby enhancing the performance of the speech recognition/understanding system.

While we did see an improvement in word-level recognition scores for single pronunciation word models that used context-dependent duration modeling, we did not observe any improvement for multiple-pronunciation models. A possible explanation for this is that for the single-pronunciation word models, the context-dependent durations provide a means for capturing some of the variation that is present in the pronunciations of the words. Once this variation is modeled explicitly, as in the multiple-pronunciation word models, there is no more benefit to detailed duration modeling. Another possible explanation may be that the conservative approximation that was used when choosing a duration for a phoneme that had multiple contexts was a poor match for the data.

Additionally, we found that the semantic scores for those systems that used context-dependent duration modeling were slightly worse than for the systems that used context-independent duration modeling. We hypothesized that the context-dependent duration models mainly improved the recognition of semantically insignificant "function" words and therefore did not improve the semantic scores. A preliminary examination of this hypothesis showed that the ten most frequently misrecognized words in both the single-pronunciation and multiple-pronunciation systems were function words which generally have little effect on the semantics of the sentence.

These findings illustrate an important difference between speech recognition systems and speech understanding systems. If it is important to recognize every word that a person says, such as in an automatic dictation system, then a significant amount of effort must go into modeling the words which cause the most errors – function words. However, if the goal is to "understand" what a person says, then the function words are less important than those words that are semantically "meaningful" to the task.

5.1.2 Multiple Pronunciation Modeling

When one considers the variety of realizations that a word may have depending on its phonological context, the dialect of the speaker, etc., it is clear that models which allow only a single pronunciation for a word cannot accurately characterize the word's pronunciation. This was demonstrated in our system by a reduction in the word-level error rate of over 20% when replacing single-pronunciation word models with multiple-pronunciation word models. Also, while the multiple-pronunciation word models improved the word-level performance, they also improved the performance of the system at the semantic level. This indicates that the multiple-pronunciation word models are important for the recognition of the semantically meaningful words in addition to the more frequently occurring function words.

There are many difficulties associated with the construction of a model that attempts to capture linguistic variation in words. Such difficulties include: how does one derive alternate pronunciations for a word; how can the fact that certain pronunciations are more likely than others be represented; and how can the pronunciations be tailored to a particular

speech recognizer?

In this dissertation, we have presented an approach which attempts to overcome some of the difficulties associated with the development of multiple-pronunciation word models. The technique we have proposed integrates well with the embedded training procedure that has been used for the training of Multi-layer Perceptrons for continuous speech recognition (Bourlard & Wellekens 1989; Bourlard & Morgan 1993). At each step in the embedded training procedure, the probabilities of alternative pronunciations for a word are reestimated using the HMM merging algorithm (Stolcke & Omohundro 1993b). The HMM merging algorithm not only reestimates transition probabilities, but it induces the topology of the model allowing the model to generalize, thus enabling it to produce pronunciations that it may not have observed.

We found a significant improvement at both the word level and the semantic level through the use of multiple-pronunciation word models developed with this approach.

Although we have a relatively small vocabulary for the BeRP system, we expect these results to carry over to large vocabulary systems such as those used in the *Wall Street Journal* task. Such large systems may benefit from the use of this technique even more than the BeRP system because of the increase in the amount of training data that would be available for estimating the parameters of the multiple-pronunciation models.

5.1.3 Loosephones

In Section 4.4 we discussed how a system that is built using dictionary pronunciations is constrained by the linguistic knowledge that has gone into the creation of the pronunciations. It was hypothesized that if one could loosen these constraints, the system would be freeer to find a better match between the MLP and the word models, resulting in an increase in recognition performance. The results obtained in the Loosephones experiments reported in Section 4.5.2 did not support this hypothesis. In fact, these results would seem to indicate that making use of this type of linguistic knowledge is beneficial (at least for the BeRP task).

It is possible that because of the relatively small size of our training corpus, we could not reliably estimate the parameters of the large word models that are created in the Loosephones algorithm. We are still hopeful that given enough training data, this approach may yield improvements.

Another possible reason that we did not observe an improvement through Loosephones may be that we did not iterate through the process enough times. At each iteration, the MLP must be retrained, which is a very time-consuming process (roughly 13 hours on the RAP machine).

A potentially useful application for this algorithm may be to use the Loosephones pronunciations for those words for which it is difficult to obtain pronunciations from other sources. This could be accomplished by performing phone recognition on the word and using the sequence of recognized phones as the pronunciation for the word. In fact, it may be useful to add the most-likely pronunciations for each of the Loosephones word models

to the initial source of pronunciations as part of the standard multiple-pronunciation word model construction algorithm. This would allow for pronunciations of words as "seen" by the MLP, while at the same time making use of linguistic knowledge.

Additionally, while the Loosephones did not give improved word-level performance over the word models that were generated from a text-to-phoneme system, they worked just as well at the semantic level. The Loosephones experiments performed in this work are very preliminary and with further experimentation, we believe that the word-level scores could be improved such that they would show improvements over the text-to-phoneme word models. If the Loosephones can be made to work better than the text-to-phoneme word models, then there will be a significant advantage to their use in constructing new speech recognition systems, especially when constructing systems for different languages.

5.1.4 Future Improvements to BeRP

The BeRP system as presented in this thesis represents a "snapshot" at one instant in time of the BeRP system. The system is continually undergoing refinement and modification. There are several improvements that are planned for BeRP. This section briefly lists some of these improvements.

Better Acoustic Training

All of the experiments reported in this thesis were performed using an MLP that was trained on roughly 2,300 utterances (700,000 frames of training data), which is a very small amount of data by today's standards. The collection and processing of utterances continues at ICSI and the new data will be used to retrain the MLPs. Another possibility for increasing the amount of data on which the MLP is trained is to initialize the BeRP MLP from an MLP that was trained on a large amount of data, such as the 5,000,000 frames of data in the WSJ0 corpus (Paul & Baker 1992).

Better Language Modeling

The bigram language model we currently use in BeRP was calculated from the stochastic context-free grammar (SCFG) (described in Section 3.4) by using the SCFG to generate several thousand sentences and then estimating the bigram probabilities from those sentences. The SCFG currently only covers approximately 65% of the training sentences, and thus there are a large number of possible bigram pairs that will not be allowed by the current bigram grammar. In fact, out of the 364 utterances in the test set used for this thesis, 137 had out-of-grammar (OOG) bigram pairs. The OOG bigram pairs are pairs of words for which there are no probabilities in the bigram grammar.

If we test on the 227 utterances that have no OOG bigram pairs, the word-level error rate for the multiple-pronunciation models with context-dependent durations drops from 32.1% to 19.8% – a reduction of 38.3%. The 32.1% error rate corresponds to 709 misrecognized words out of the 2,208 words in the test utterances. Out of these 709 misrecognized words,

only 248 (35%) occur in the set of 227 utterances that have no OOG bigram pairs. Thus, the majority of the misrecognized words are in the smaller set of 137 utterances that have OOG bigram pairs. The error rate on these utterances is 48.3%, which represents 461 (65%) of the 709 misrecognized words.

Excluding the grammatical constraints of English, there are two possible causes for OOG bigram pairs: out-of-vocabulary words and a lack of data. We can separate the effects of OOV words from the effects of insufficient data by splitting the set of 137 utterances into two sets of utterances and testing each set separately.

The first set of utterances have OOG bigrams because one or both of the words in the bigram pair is not in the vocabulary. There are 47 utterances in this first set and the error rate for these utterances is 57.0%. This represents 176 (38%) of the 461 misrecognized words in the OOG sentences. The second set of utterances have OOG bigrams because the particular sequence of (in-vocabulary) words was not observed during the construction of the bigram grammar. There are 90 utterances in this set and the error rate is 44.2%, which represents 285 (62%) of the 461 misrecognized words in the OOG sentences.

These results suggest two changes that should significantly improve the performance of BeRP. The first change is to add more words to the vocabulary, an easy change to implement. The second change is to apply some form of smoothing to the bigram grammar to account for unobserved bigram pairs. While smoothing the bigram is not as simple as just adding words to the vocabulary, it should provide the largest reduction in the error rate given the analysis presented above.

Dynamic Grammars

Currently, the BeRP system does not assume that a user's response will be related to its prompts. However, we have noticed that the majority of users do respond appropriately to the prompts from the system. We can take advantage of this by altering the probabilities in the grammar depending on the current prompt. For example, if the system asks the user "What type of food would you like to eat?", the probability for the various food types should be increased while the probabilities for words that are less likely, given the current prompt, should be decreased.

Current Research Areas

The possible improvements to BeRP mentioned in the previous sections are only incremental points. There are several, more interesting research issues that are currently being investigated at ICSI. These include gender modeling, modeling temporal dynamics, tightly-coupled stochastic context-free grammars, accent detection, and improving phoneme probability estimates by combining the estimates from several MLPs.

5.2 Limitations

While the BeRP task is a difficult task, it is relatively small by today's standards. As mentioned in Chapter 1, modern systems are beginning to use vocabularies of up to 20,000 words and roughly 40,000 utterances of training data. Although the techniques proposed in this thesis have worked well for BeRP, it is not known whether they will provide as much improvement in a larger system.

Additionally, the BeRP task is not a "standard" on which other researchers have tested. Thus, there are no comparative results for this task with which to gauge our performance. The word model construction techniques proposed in this thesis should be tested in a more standard task (e.g. ATIS [Price 1990]) to determine whether these results are robust.

5.3 Conclusion

There are many sources of variability that contribute to the difficulty of automatic speech recognition/understanding. We humans have incredibly robust mechanisms that we use to overcome the uncertainty found in our environment. It is hoped that through our attempts to model this robustness, we may begin to understand these mechanisms.

Bibliography

BAHL, L. R., R. BAKIS, P. S. COHEN, A. G. COLE, F. JELINEK, B. L. LEWIS, & R. L. MERCER. 1980. Further results on the recognition of a continuously read natural corpus. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*.

BAKER, J. K., 1975. *Stochastic modeling as a means of automatic speech recognition*. Carnegie Mellon University dissertation.

BAKIS, R. 1976. Continuous speech recognition via centisecond acoustic states. In *91st Meeting Acoustical Society of America*, Washington, DC.

BAUM, L. E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3.1–8.

——, & T. PETRIE. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 36.1554–1563.

——, T. PITRIE, G. SOULES, & N. WEISS. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions in Markov chains. *The Annals of Mathematical Statistics* 41.164–171.

BERGER, J. O. 1985. *Statistical decision theory and Bayesian analysis*. New York: Springer Verlag.

BOURLARD, H., J. BOITE, B. D'HOORE, & M. SAERENS. 1993. Performance comparison of Hidden Markov Models and neural networks for task dependent and independent isolated word recognition. In *Proceedings European Conf. on Speech Communication and Technology. (EUROSPEECH)*, 1925–1928, Berlin, Germany.

——, & N. MORGAN. 1991. Merging Multilayer Perceptrons & Hidden Markov Models: Some experiments in continuous speech recognition. In *Artificial Neural Networks: Advances and Applications*, ed. by E. Gelenbe. North Holland Press.

——, & N. MORGAN. 1993. *Connectionist speech recognition a hybrid approach*. Kluwer Academic Publishers.

——, N. MORGAN, & C. WOOTERS. 1991. Connectionist approaches to the use of Markov Models for speech recognition. In *Advances in Neural Information Processing Systems*, ed. by D. S. Touretzky & R. Lippman, volume 3, San Mateo. Morgan Kaufmann.

——, & C. J. WELLEKENS. 1989. Links between Markov models and multilayer perceptrons. In *Advances in Neural Information Processing Systems 1*, ed. by D.J. Touretzky, 502–510, San Mateo. Morgan Kaufmann.

——, & C. J. WELLEKENS. 1990. Links between Markov models and multilayer perceptrons. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12.1167–1178.

BRIDLE, J. S. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, architectures and applications*, ed. by F. Fogelman Soulie & J. Herault, NATO ASI, 227–236.

BUTZBERGER, J., H. MURVEIT, E. SHRIBERG, & P. PRICE. 1992. Spontaneous speech effects in large vocabulary speech recognition applications. In *Proceedings DARPA Speech and Natural Language Workshop*, 339–343.

CHIGIER, B., & H. C. LEUNG. 1992. The effects of signal representations, phonetic classification techniques, and the telephone network. In *Proceedings Int'l. Conf. on Spoken Language Precessing*, volume 1, 97–100, Banff, Canada.

CHOW, Y. L., M. O. DUNHAM, O. KIMBALL, M. KRASNER, F. KUBALA, J. MAKHOUL, S. ROUCOS, & R. M. SCHWARTZ. 1987. BYBLOS: the BBN continuous speech recognition system. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 89–92.

——, R. M. SCHWARTZ, S. ROUCOS, O. KIMBALL, P. PRICE, F. KUBALA, M. DUNHAM, M. KRASNER, & J. MAKHOUL. 1986. The role of word-dependent coarticulatory effects in a phoneme-based speech recognition system. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*.

COHEN, M., 1989. *Phonological structures for speech recognition*. University of California, Berkeley dissertation.

——, H. FRANCO, N. MORGAN, D. RUMELHART, & V. ABRASH. 1992. Hybrid Neural Network/Hidden Markov Model continuous speech recognition. In *Proc. Int'l Conf. on Spoken Lang. Processing*, Banff, Canada.

COHEN, P. S., & P. L. MERCER. 1975. The phonological component of an automatic speech recognition system. In *Speech recognition*, ed. by R. Reddy, 275–320. New York: Academic Press.

DARPA. 1992. *Proceedings of the DARPA speech and natural language workshop*.

DAVIS, K. H., R. BIDDULPH, & S. BALASHEK. 1952. Automatic recognition of spoken digits. *JASA* 24.637–642.

DAVIS, S. B., & P. MERMELSTEIN. 1980. Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.357–366.

DEMPSTER, A. P., N. M. LAIRD, & D. B. RUBIN. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B* 34.1–38.

EDWARDS, HAROLD T. 1992. *Applied phonetics: The sounds of American English*. Singular Publishing Group Inc.

FERGUSON, J. D. 1980. Variable duration models for speech. In *Symposium on the Application of Hidden Markov Models to Text and Speech*, 143–179.

FRAZER, N. M., & G. N. GILBERT. 1991. Simulating speech systems. *Computer Speech and Language* 5.81–99.

G. D. FORNEY, JR. 1973. The Viterbi algorithm. *Proc. IEEE* 61.268–78.

GOODINE, D., S. SENEFF, L. HIRSCHMAN, & M. PHILLIPS. 1991. Full integration of speech and language understanding in the MIT spoken language system. In *Proceedings of Eurospeech 91*, 24—26, Genova, Italy.

HERMANSKY, H. 1990. Perceptual linear predictive (PLP) analysis of speech. *JASA* 87.

——, & N. MORGAN. 1992. Towards handling the acoustic environment in spoken language processing. In *Proceedings ICSLP*, volume 1, 85–88, Banff, Alberta, Canada.

——, N. MORGAN, A. BAYYA, & P. KOHN. 1991. Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP). *Proceedings European Conf. on Speech Communication and Technology. (EUROSPEECH)* 1367–1370.

——, N. MORGAN, A. BAYYA, & P. KOHN. 1992. RASTA-PLP speech analysis technique. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 1, 121–124, San Francisco, CA. IEEE.

——, N. MORGAN, & H. G. HIRSCH. 1993. Recognition of speech in additive and convolutional noise based on RASTA spectral processing. In *Proceedings Int'l Conf. on Acoustics Speech and Signal Processing*, volume II, 83–86, Minneapolis, Minnesota, USA. IEEE.

HIRSCH, H. G., 1993. Personal Communication.

——, P. MEYER, & H. W. RUEHL. 1991. Improved speech recognition using high-pass filtering of subband envelopes. In *Proceedings European Conf. on Speech Communication and Technology. (EUROSPEECH)*, 413–416, Genoa.

HOCHBERG, M., & H. SILVERMAN. 1993. Constraining model duration variance in HMM-based connected-speech recognition. In *Proceedings Eurospeech*.

JELINEK, F. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE* 64.532–556.

——, & R. L. MERCER. 1980. Interpolated estimation of markov source parameters from sparse data. In *Pattern recognition in practice*, ed. by E. S. Gelsema & L. N. Kanal, 381–397. Amsterdam, The Netherlands: North-Holland Publishing Company.

JUANG, B. H., & L. R. RABINER. 1990. The segmental K-Means algorithm for estimating parameters of Hidden Markov Models. *IEEE Trans. on Acoustics, Speech, and Signal Processing* 38.1639–41.

JURAFSKY, D. 1992b. An on-line computational model of human sentence interpretation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 302–308, San Jose, CA.

——, 1992a. *An on-line computational model of human sentence interpretation: A theory of the representation and use of linguistic knowledge*. Berkeley, CA: University of California at Berkeley dissertation.

——, C. WOOTERS, G. TAJCHMAN, & N. MORGAN. 1993. The Berkeley Restaurant Project: A status report at phase I. Technical report, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704. to appear.

KAY, M. 1973. The MIND system. In *Natural language processing*, ed. by Randall Rustin, 155–188. New York: Algorithmics Press.

KLATT, D. H. 1975. Voice onset time, friction, and aspiration in word-initial consonant clusters. *Journal of Speech and Hearing Research* 18.686–706.

KUBALA, G. F., Y. CHOW, A. DERR, M. FENG, O. KIMBALL, J. MAKHOUL, P. PRICE, J. ROHLICEK, S. ROUCOS, R. SCHWARTZ, & J. VANDEGRIFT. 1988. Continuous speech recognition results of the BYBLOS system on the DARPA 1000-word Resource Management databasement. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 291–294.

KUCHERA, H., & W. N. FRANCIS. 1967. *Computational analysis of present-day American English*. Providence, Rhode Island: Brown University Press.

LEE, K. F. 1989. *Automatic speech recognition: The development of the SPHINX system*. Kluwer Academic Publishers.

LEUNG, H. C., & V. W. ZUE. 1984. A procedure for automatic alignment of phonetic transcriptions with continuous speech. In *Proc. International Conference on Acoustics Speech and Signal Processing*, 2.7.1–2.7.4. IEEE.

LEVINSON, S. E. 1986. Continuously variable duration Hidden Markov Models for automatic speech recognition. *Computer Speech and Language* 29–45.

——, L. R. RABINER, & M. M. SONDHI. 1983. An introduction to the application of the theory of probabilistic functions on a Markov process to automatic speech recognition. *Bell system Technical Journal* 62.

MACKAY, D. J. C. 1992. Bayesian interpolation. *Neural Computation* 4.415–447.

MOORE, R., F. PEREIRA, & H. MURVEIT. 1989. Integrating speech and natural-language processing. In *Proceedings DARPA Speech and Natural Language Workshop*, 243–247.

MOORE, R. K., & A. MORRIS. 1992. Experiences collecting genuine spoken enquiries using WOZ techniques. In *Proceedings DARPA Speech and Natural Language Workshop*, Harriman, New York.

——, M. J. TOMLINSON, & A. MORRIS. 1991. Whither the wizard? In *Proc. ESCA Workshop on the Structure of Multimodal Dialogue*, Maratea, Italy.

MORGAN, N., J. BECK, P. KOHN, J. BILMES, E. ALLMAN, & J. BEER. 1992. The Ring Array Processor (RAP): a multiprocessing peripheral for connectionist applications. *Journal of Parallel and Distributed Computing* 248–259.

——, & H. BOURLARD. 1990. Continuous speech recognition using Multilayer Perceptrons with Hidden Markov Models. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 413–416, Albuquerque, New Mexico. IEEE.

——, H. HERMANSKY, H. BOURLARD, P. KOHN, & C. WOOTERS. 1991a. Phonetically-based speaker independent continuous speech recognition using PLP analysis with multilayer perceptrons. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Toronto, Canada. IEEE.

——, C. WOOTERS, H. BOURLARD, & M. COHEN. 1990. Continuous speech recognition on the resource management database using connectionist probability estimation. In *Proceedings 1990 International Conference on Spoken Language Processing*, Kobe, Japan. Acoustical Society of Japan.

——, C. WOOTERS, & H. HERMANSKY. 1991b. Experiments with temporal resolution for continuous speech recognition with multi-layer perceptrons. In *Proceedings IEEE Workshop on Neural Networks for Signal Processing*, ed. by Candace A. Kamm B.H. Juang, S.Y. Kung. IEEE.

MURVEIT, H., J. BUTZBURGER, & M. WEINTRAUB. 1992. Reduced channel dependence for speech recognition. In *Proceedings DARPA Speech and Natural Language Workshop*.

——, & M. WEINTRAUB. 1988. 1000-word speaker independent continuous-speech recognition using Hidden Markov Models. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 115–118.

PAUL, D. B., & J. M. BAKER. 1992. The design for the Wall Street Journal-based CSR corpus. In *Proc. Fifth DARPA Speech and Natural Language Workshop*, 357–362.

PIERACCINI, R., & A. E. ROSENBERG. 1989. Automatic generation of phonetic units for continuous speech recognition. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 1, 623–6, Glasgow, Scotland.

PITCHER, D. 1989. *Berkeley City Guide*. Heyday Books.

PRICE, P. 1990. Evaluation of spoken language systems: The ATIS domain. In *Proc. Third DARPA Speech and Language Workshop*, 91–95, Hidden Valey, PA.

——, W. FISHER, J. BERNSTEIN, & D. PALLET. 1988. The darpa 1000-word resource management database for continuous speech recognition. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, 651–654, New York. IEEE.

RABINER, L. R., & B. H. JUANG. 1986. An introduction to Hidden Markov Models. *IEEE ASSP Magazine* 3.4–16.

RENALS, S., N. MORGAN, H. BOURLARD, M. COHEN, H. FRANCO, C. WOOTERS, & P. KOHN. 1991. Connectionist speech recognition: Status and prospects. Technical Report TR-91-070, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704.

ROBINSON, A. J., L. ALMEIDA, J.-M. BOITE, H. BOURLARD, F. FALLSIDE, M. HOCHBERG, D. KERSHAW, P. KOHN, Y. KONIG, N. MORGAN, J. P. NETO, S. RENALS, M. SAERENS, & C. WOOTERS. 1993. A neural network based, speaker independent, large vocabulary, continuous speech recognition system: The WERNICKE project. In *Proceedings European Conf. on Speech Communication and Technology. (EUROSPEECH)*, 1941–1944, Berlin, Germany.

RUMELHART, D. E., G. E. HINTON, & R. J. WILLIAMS. 1986. Learning internal representations by error propagation. In *Parallel distributed processing. explorations of the microstructure of cognition*, ed. by D. E. Rumelhart & J. L. McClelland, volume 1: Foundations. MIT Press.

RUSSELL, M. J., & R. K. MOORE. 1985. Explicit modelling of state occupancy in Hidden Markov Models for automatic speech recognition. In *Proceedings ICASSP*, 5–8.

SCHWARTZ, R. M., Y. L. CHOW, O. KIMBALL, S. ROUCOS, M. KRASNER, & J. MAKHOUL. 1985. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*.

——, Y. L. CHOW, S. ROUCOS, M. KRASNER, & J. MAKHOUL. 1984. Improved Hidden Markov Modeling of phonemes for continuous speech recognition. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*.

SENEFF, S., H. MENG, & V. ZUE. 1992. Language modelling for recognition and understanding using layered bigrams. In *Proceedings Int'l Conference on Spoken Language Processing*, I.317—320, Banff, Alberta, Canada.

——, & V. ZUE, 1988. Transcription and alignment of the TIMIT database. Distributed with the TIMIT database.

SOLLA, S. A., E. LEVIN, & M. FLEISHER. 1988. Accelerated learning in layered neural networks. *Complex Systems* 2.625–640.

STOLCKE, A., & S. OMOHUNDRO. 1993a. Best-first model merging for Hidden Markov Model induction. Technical report, International Computer Science Institute, 1947 Center St. Suite 600, Berkeley, CA.

——, & S. OMOHUNDRO. 1993b. Hidden Markov Model induction by Bayesian model merging. In *Advances in neural information processing systems 5*. San Mateo, Ca.: Morgan Kaufman.

VITERBI, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory* 13.260–269.

WANG, W. S-Y. 1971. The basis of speech. In *The learning of language*, ed. by Carroll E. Reed, chapter 7, 276–306. Appleton-Centruy-Crofts.

——. 1972. Approaches to phonology. In *Current trends in linguistics*, ed. by T. A. Sebeok. The Hague: Mouton.

WERBOS, P. J., 1974. *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* Cambridge, MA: Harvard University dissertation.

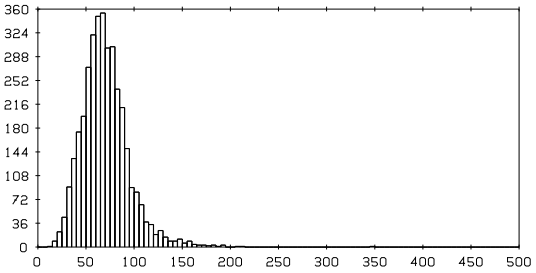
ZUE, V., J. GLASS, D. GOODINE, H. LEUNG, M. PHILLIPS, & S. SENEFF. 1990. The VOYAGER speech understanding system: Preliminary development and evaluation. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 73–76, Albuquerque, New Mexico.

ZWICKER, E. 1961. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *JASA* 33(2).248.

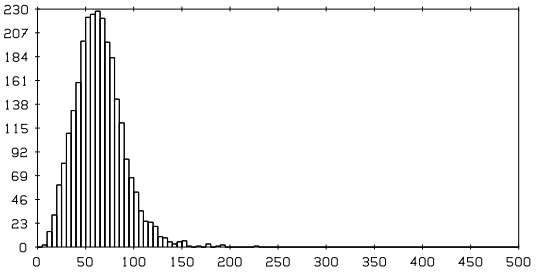
Appendix A

TIMIT Context Independent Durations

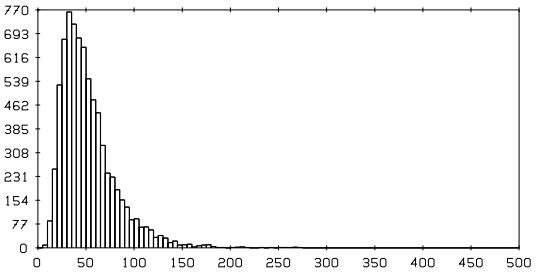
This appendix presents a set of duration histograms for each of the phones in the TIMIT database (see Section 2.2.1 for more information on the TIMIT database). Each histogram shows the distribution of the measured durations for each phone (see Table 2.3 for an explanation of the symbols). The x-axis represents observed durations, and the y-axis represents the number of times each duration was observed. Each vertical bar in a histogram represents 5 milliseconds. The TIMIT symbol, the IPA symbol, and the mean duration are shown under each histogram.



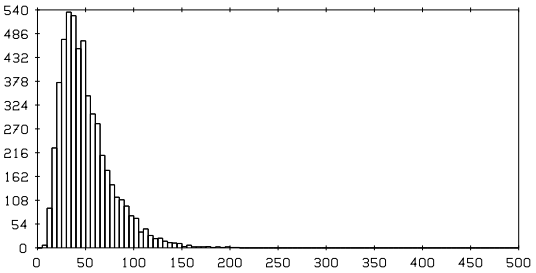
[pcl] [p^o] ($\mu = 69.62$)



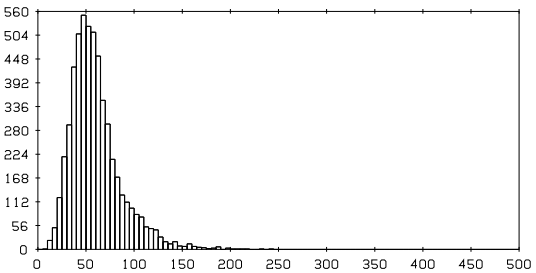
[bcl] [b^o] ($\mu = 63.70$)



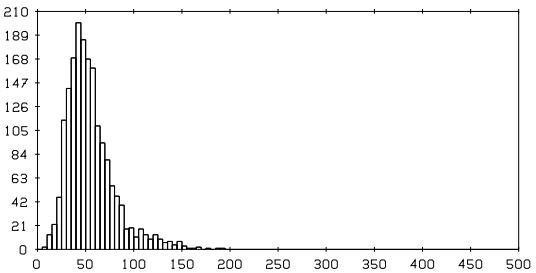
[tcl] [t^o] ($\mu = 51.98$)



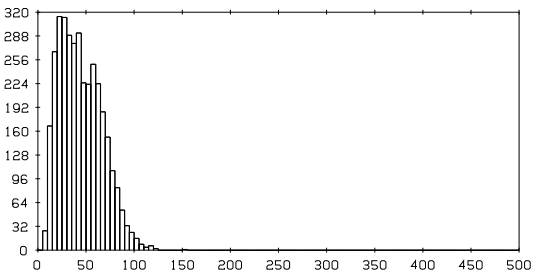
[dcl] [d^o] ($\mu = 49.95$)



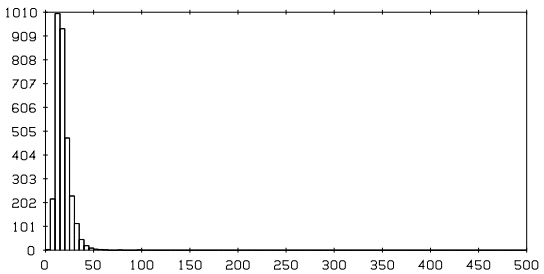
[kcl] [k^o] ($\mu = 59.67$)



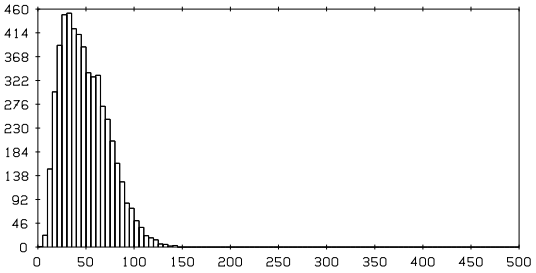
[gcl] [g^o] ($\mu = 54.60$)



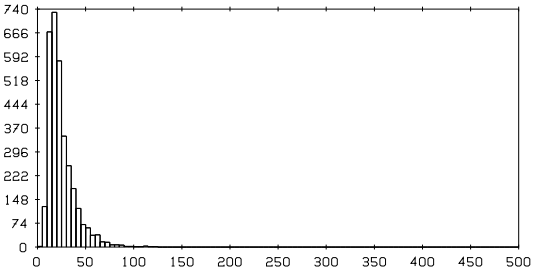
[p] ($\mu = 44.23$)



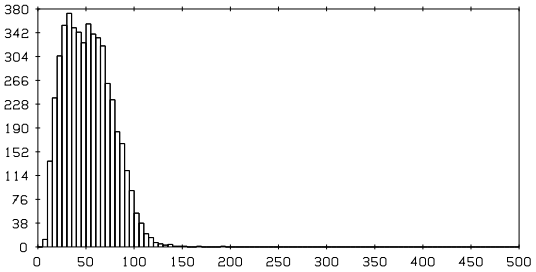
[b] ($\mu = 17.50$)



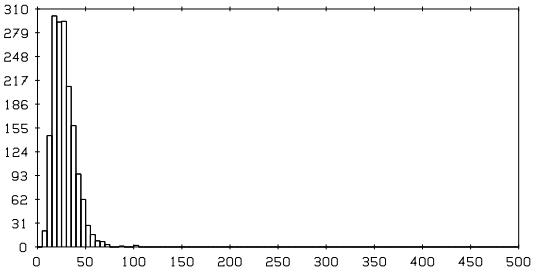
[t] ($\mu = 48.83$)



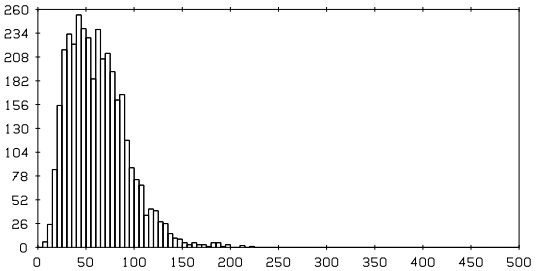
[d] ($\mu = 24.17$)



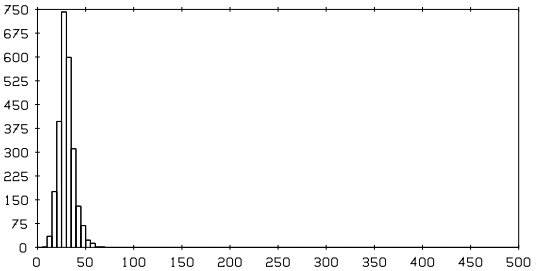
[k] ($\mu = 52.10$)



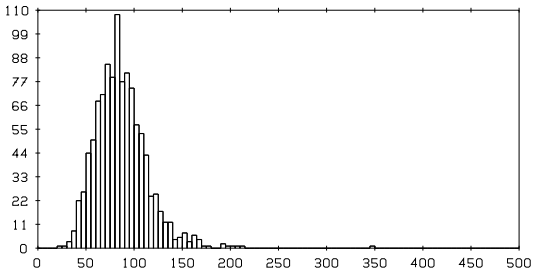
[g] ($\mu = 27.30$)



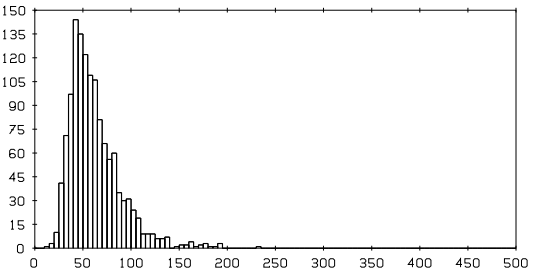
[q] [ʔ] ($\mu = 62.24$)



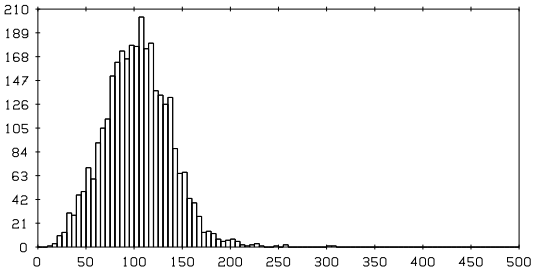
[dx] [ɾ] ($\mu = 28.81$)



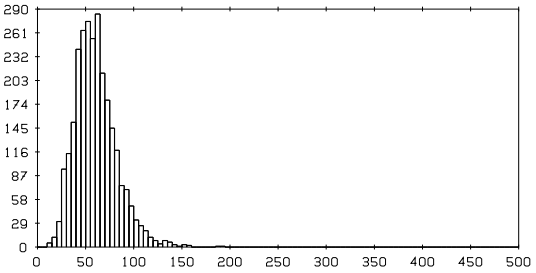
[ch] [tʃ] ($\mu = 86.36$)



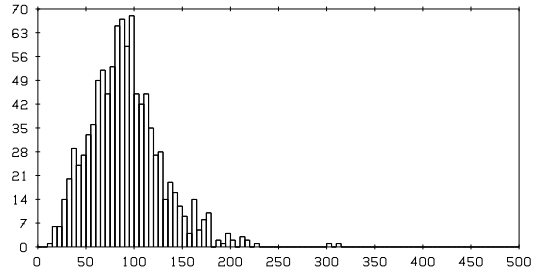
[jh] [dz] ($\mu = 61.91$)



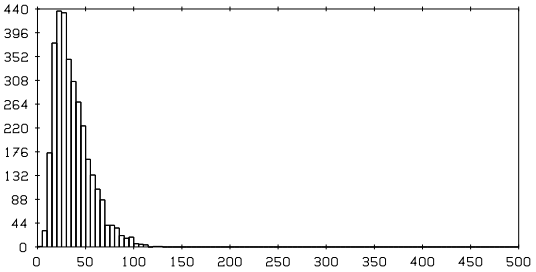
[f] (μ = 103.07)



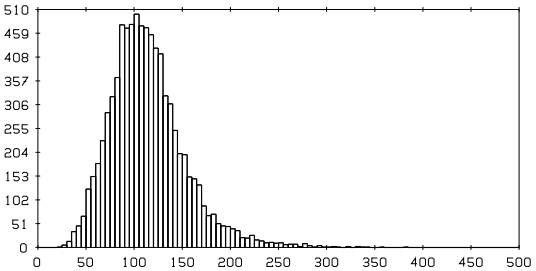
[v] (μ = 60.09)



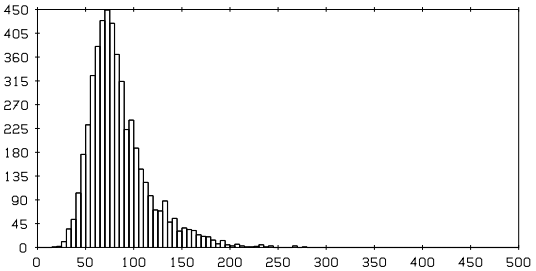
[th] [θ] (μ = 90.41)



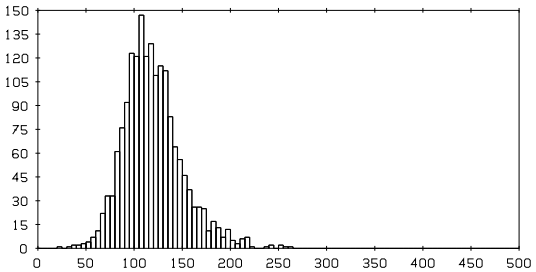
[dh] [ð] (μ = 36.24)



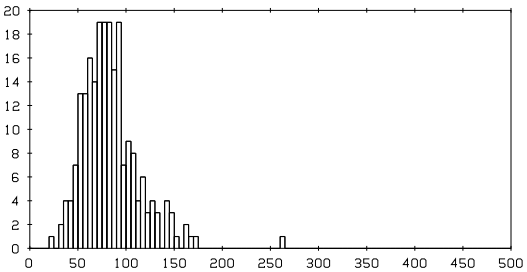
[s] (μ = 113.61)



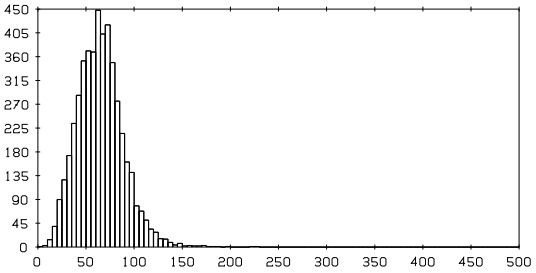
[z] (μ = 84.35)



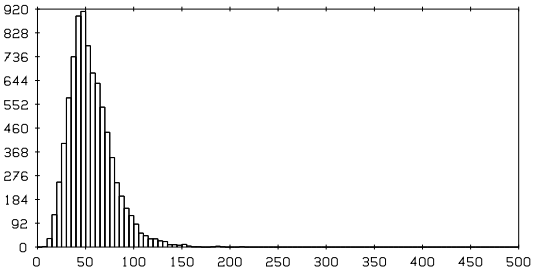
[sh] [ʃ] (μ = 118.97)



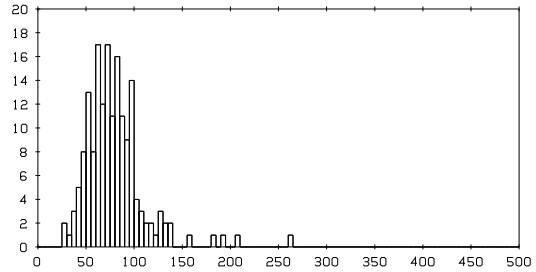
[zh] [ʒ] (μ = 83.22)



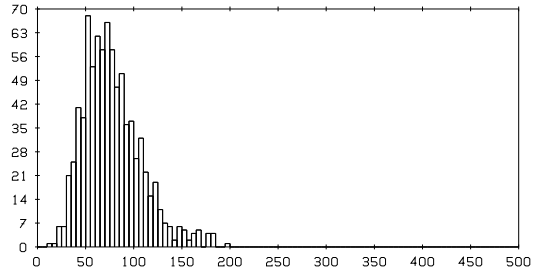
[m] ($\mu = 64.56$)



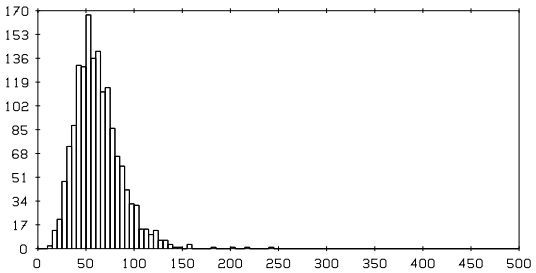
[n] ($\mu = 55.00$)



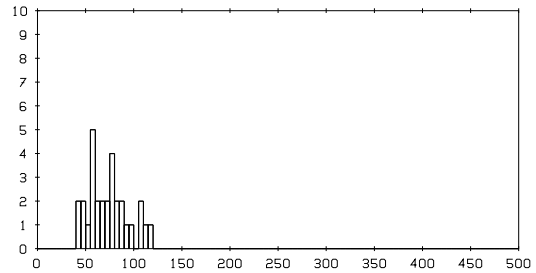
[em] [m̩] ($\mu = 79.24$)



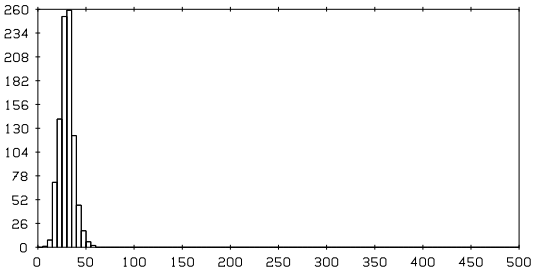
[en] [n̩] ($\mu = 77.33$)



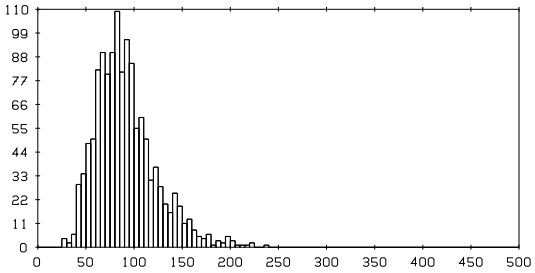
[ng] [ŋ̍] ($\mu = 61.98$)



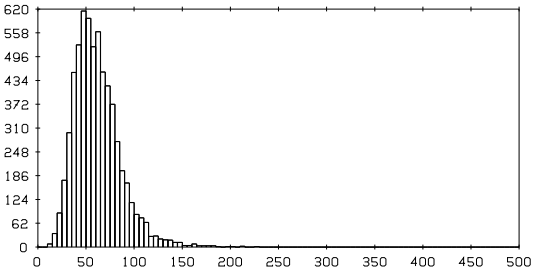
[eng] [ŋ̩] ($\mu = 73.17$)



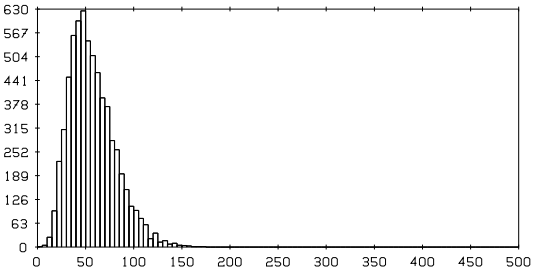
[nx] [ɾ̃] ($\mu = 28.81$)



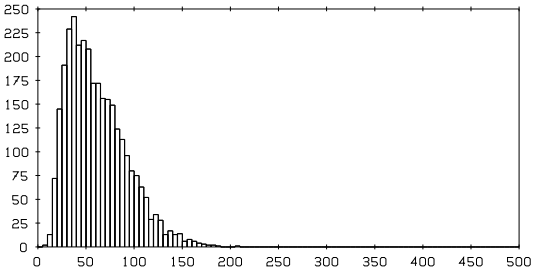
[el] [l̩] ($\mu = 91.03$)



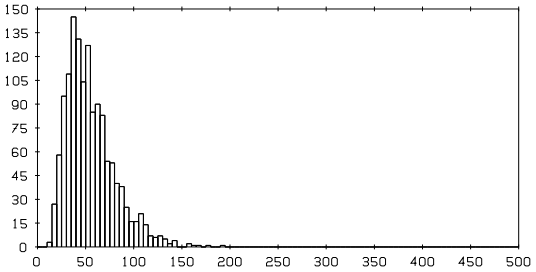
[l] (μ = 61.41)



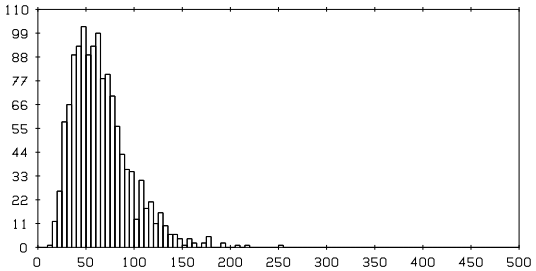
[r] (μ = 56.86)



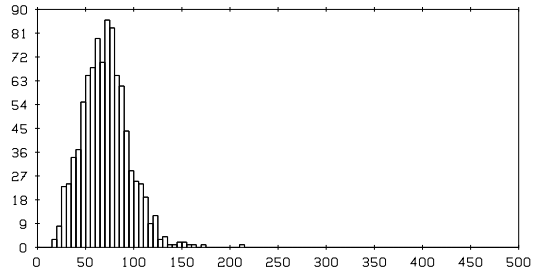
[w] (μ = 61.54)



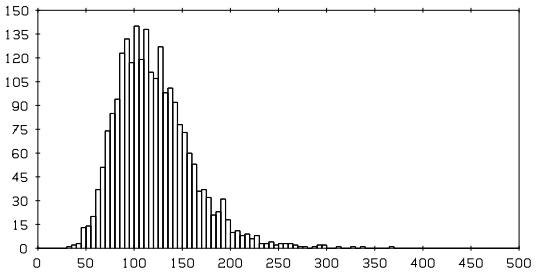
[y] (μ = 55.03)



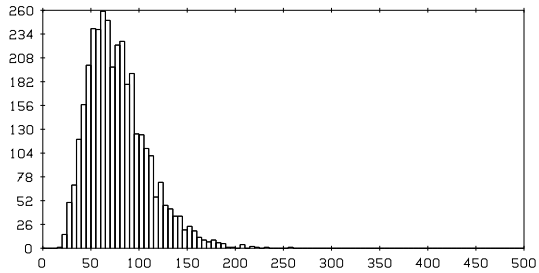
[hh] [h] (μ = 65.17)



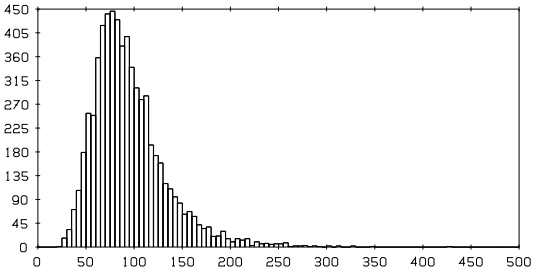
[hv] [ɦ] (μ = 70.61)



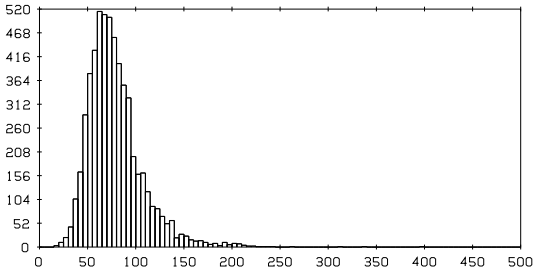
[er] [ɚ] (μ = 120.80)



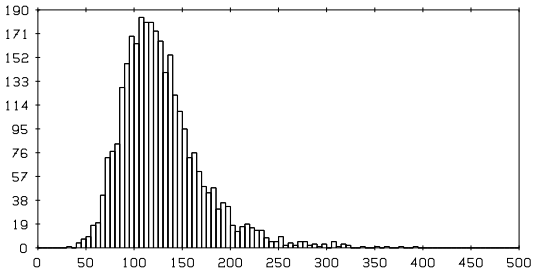
[axr] [ɝ] (μ = 77.91)



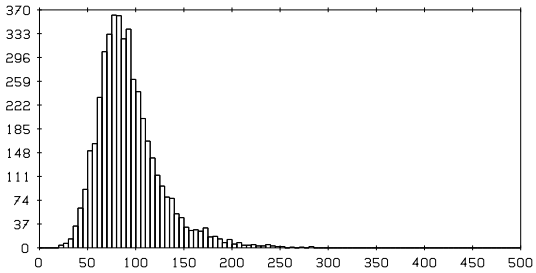
[iy] [i] (μ = 94.53)



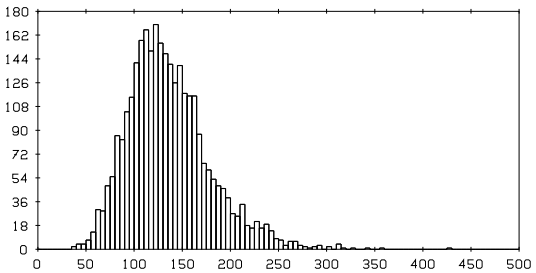
[ih] [ɪ] (μ = 78.27)



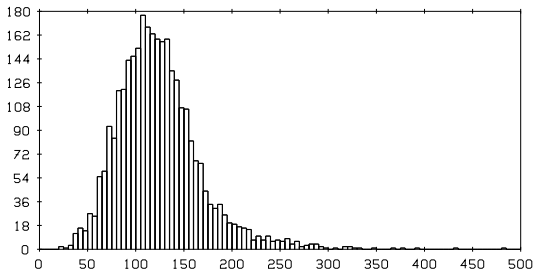
[ey] [e] (μ = 128.14)



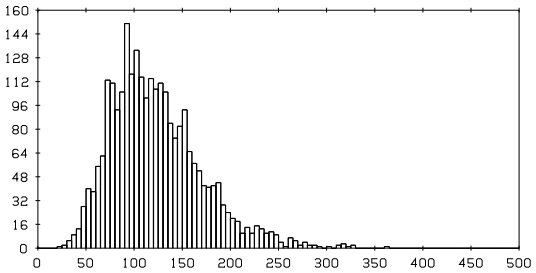
[eh] [ɛ] (μ = 92.45)



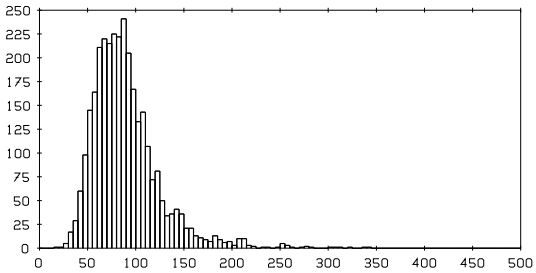
[ae] [æ] (μ = 135.83)



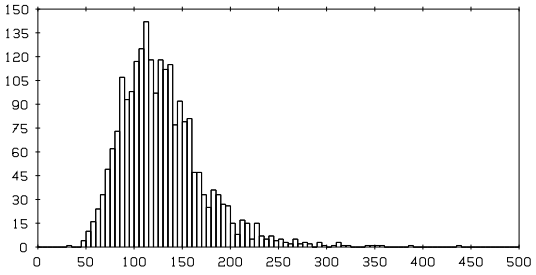
[aa] [ɑ] (μ = 123.93)



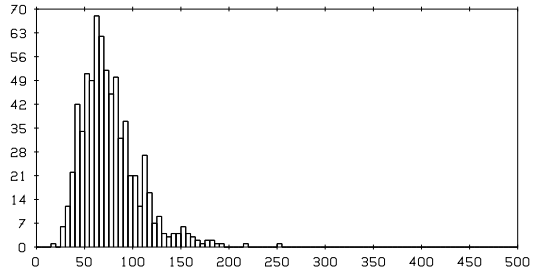
[ao] [ɔ] (μ = 122.56)



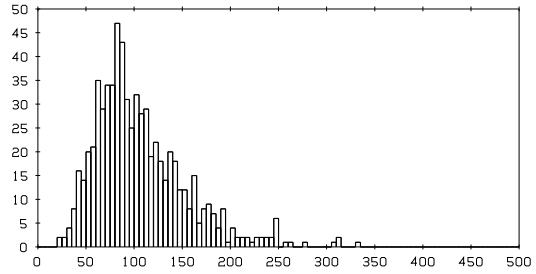
[ah] [ʌ] (μ = 89.29)



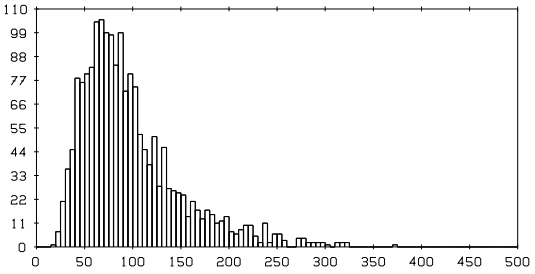
[ow] [o] ($\mu = 128.81$)



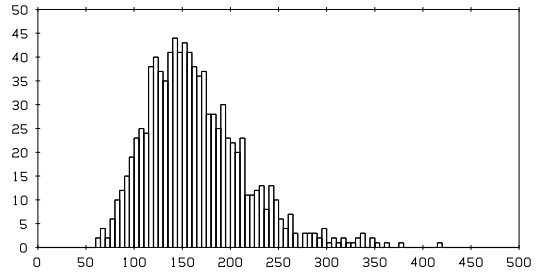
[uh] [ɷ] ($\mu = 76.44$)



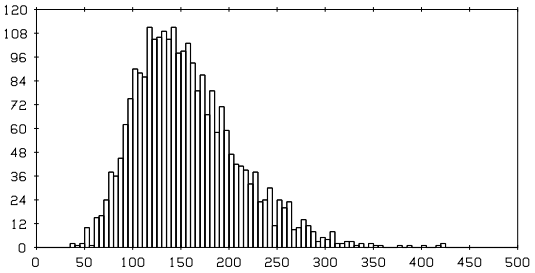
[uw] [u] ($\mu = 106.03$)



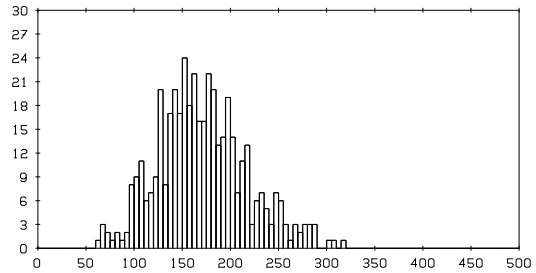
[ux] [ü] ($\mu = 97.24$)



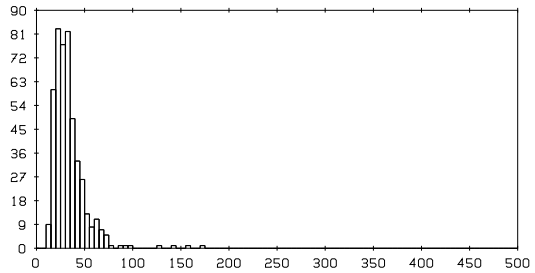
[aw] [ɑ^{w}] ($\mu = 163.60$)



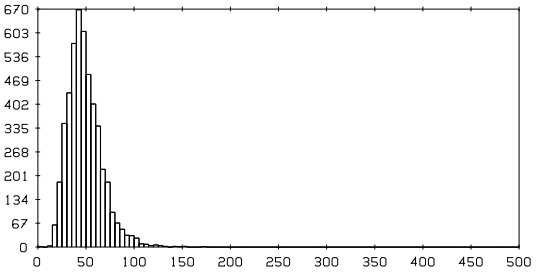
[ay] [ɑ^{y}] ($\mu = 156.07$)



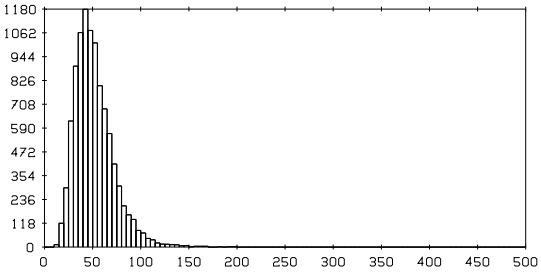
[oy] [ɔ^{y}] ($\mu = 170.91$)



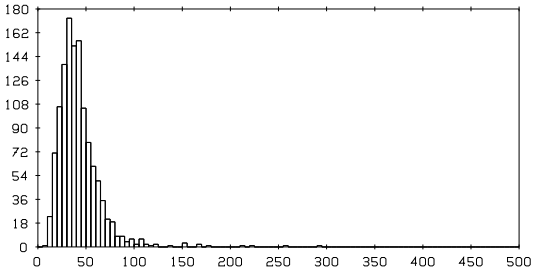
[ax-h] [ə̥] ($\mu = 33.70$)



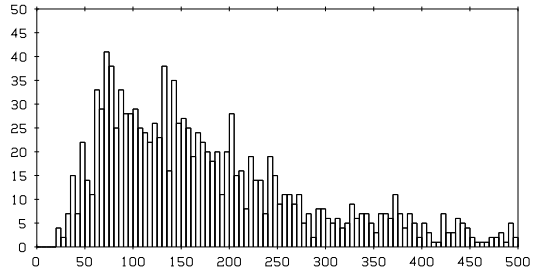
[ax] [ə] (μ = 48.43)



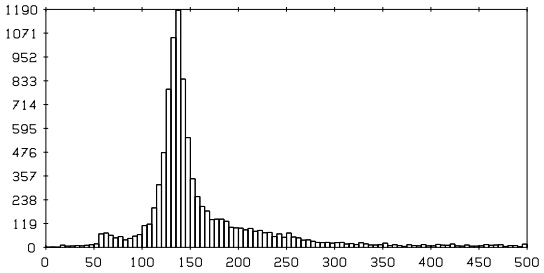
[ix] [ɨ] (μ = 51.38)



[epi] (μ = 42.42)



[pau] (μ = 189.18)



[h#] (μ = 190.57)

Appendix B

Multiple Pronunciation Word Models

This appendix contains HMM graphs for the 50 most commonly occurring words in the BeRP training corpus of 2319 utterances. These models are from the context-dependent durations experiments reported in Section 4.5. The word models were pruned using a pruning threshold of 0.25, which was found to give the best word-level scores.

In a few of the graphs there is only a single pronunciation for a word. This may be due to the pruning, and since these words are very frequent, they have been included in this appendix.

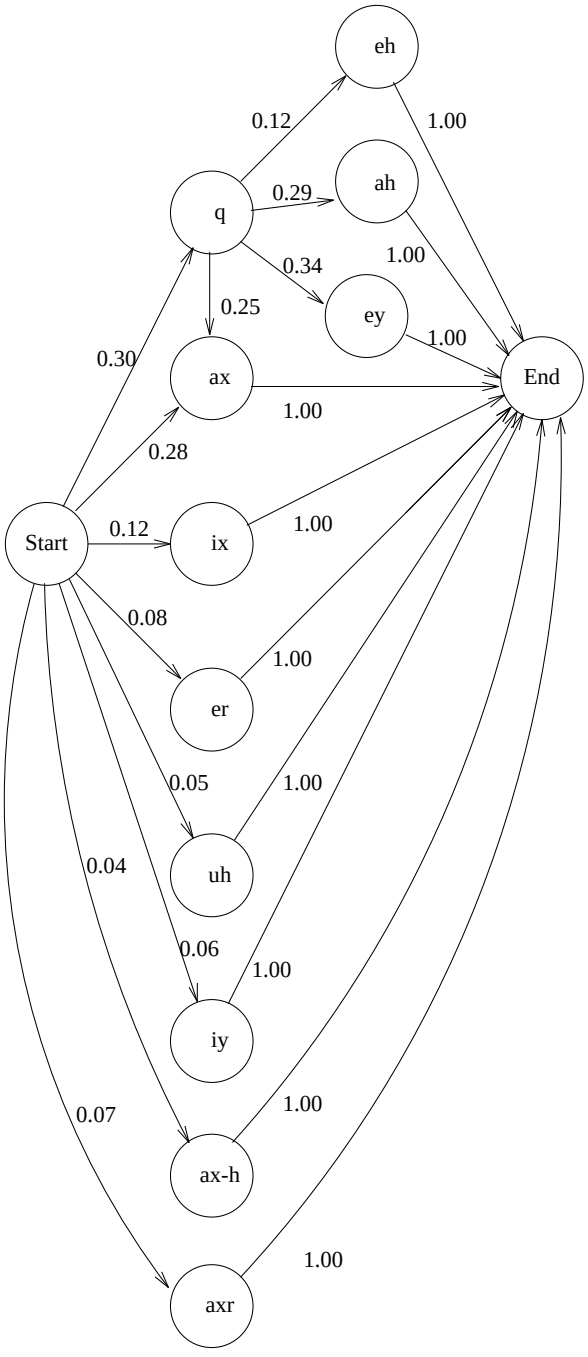

eh
0.12
1.00
q
ah
0.29
1.00
0.34
0.25
ey
1.00
0.30
ax
End
1.00
0.28
Start
0.12
ix
1.00
0.08
1.00
er
0.05
1.00
uh
0.04
0.06
1.00
iy
0.07
1.00
ax-h
1.00
axr

Figure B.1: “a”

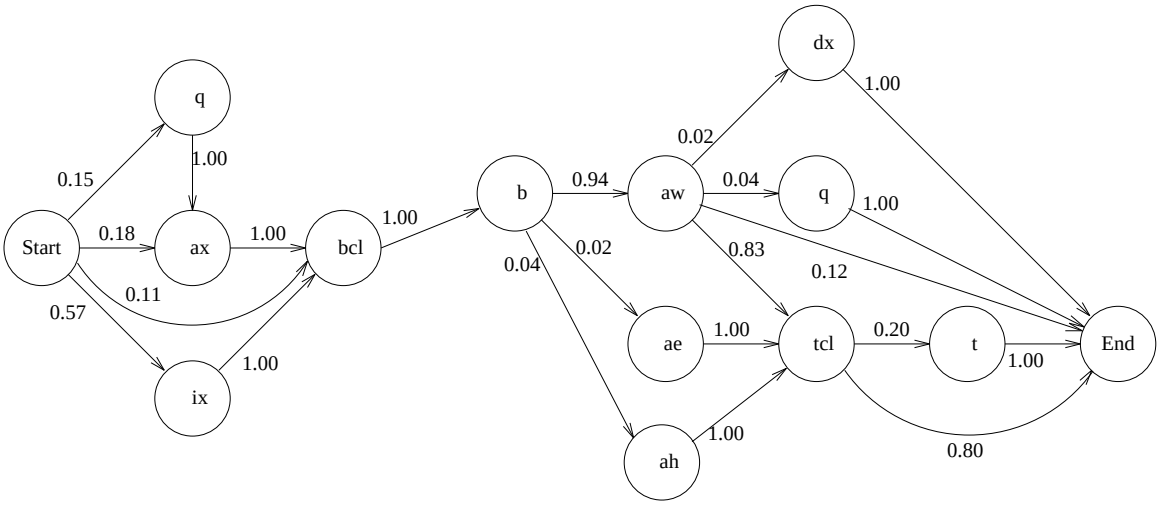


Figure B.2: “about”

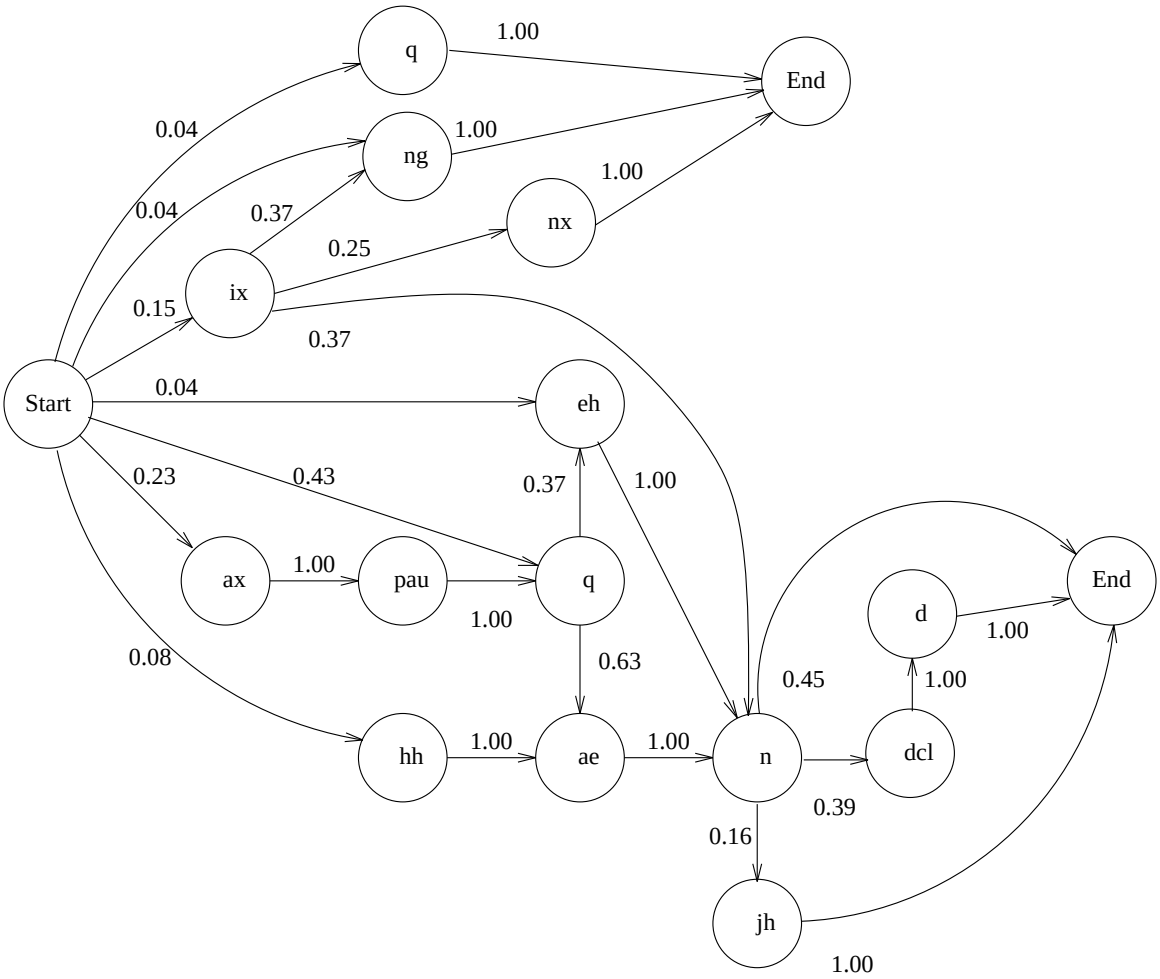


Figure B.3: “and”

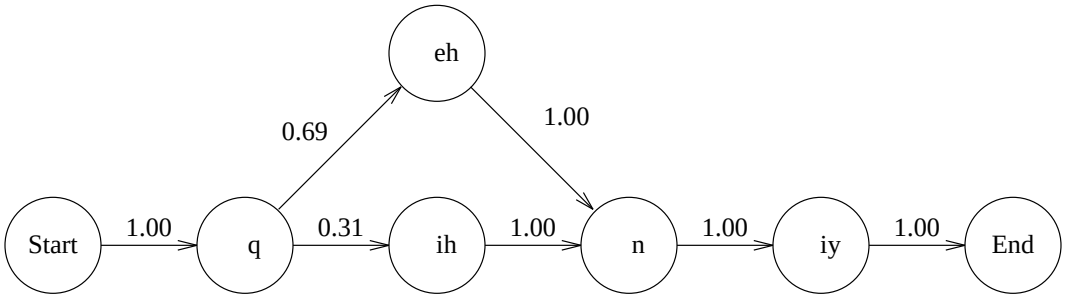


Figure B.4: “any”

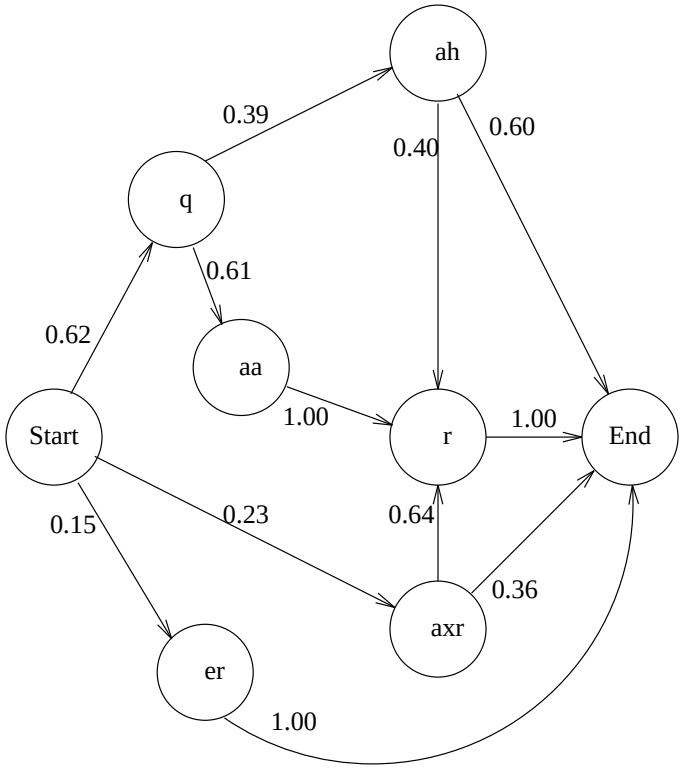


Figure B.5: “are”

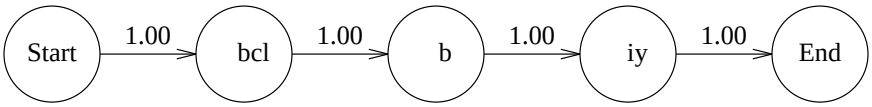


Figure B.6: “be”

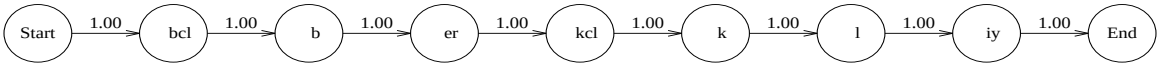


Figure B.7: “berkeley”

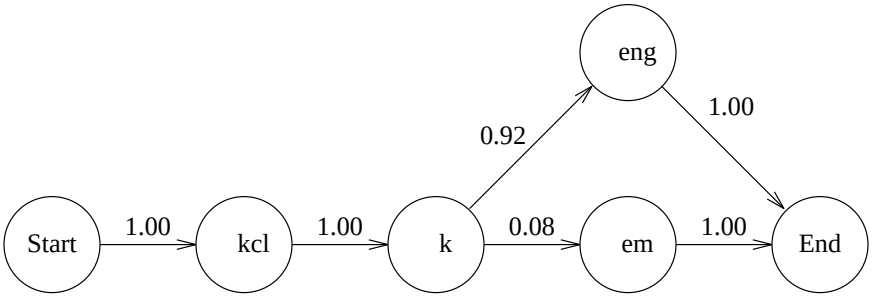


Figure B.8: “can”

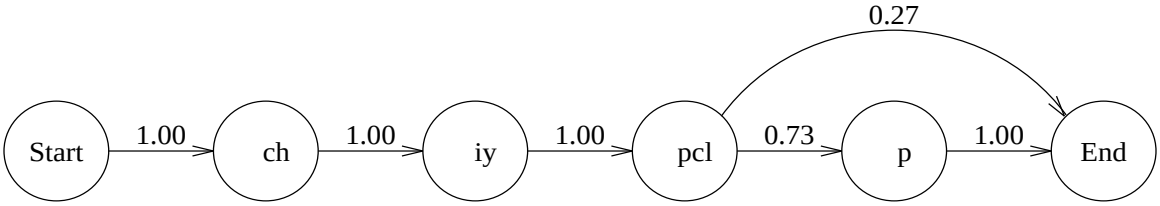


Figure B.9: “cheap”

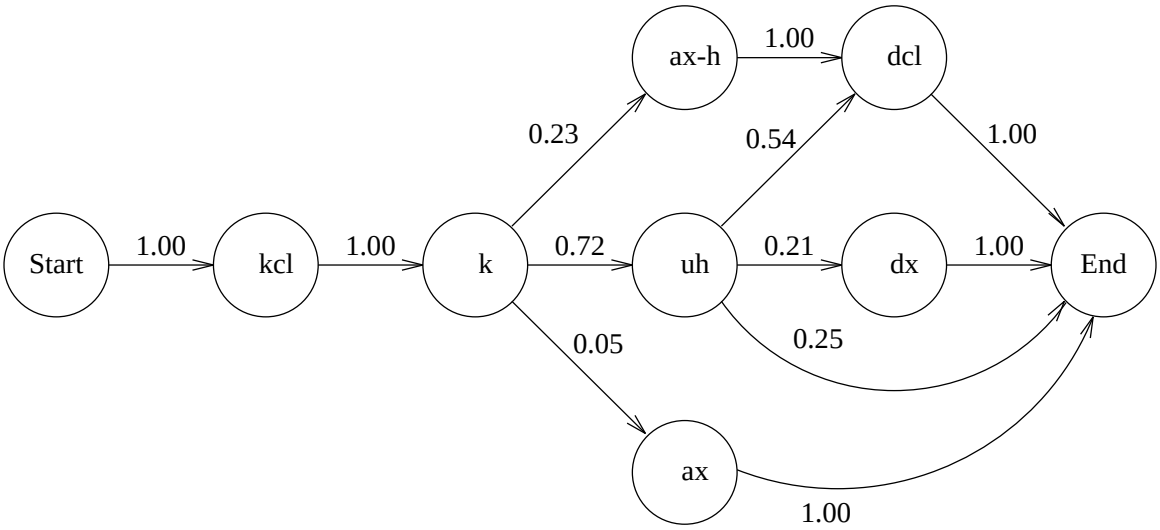


Figure B.10: “could”

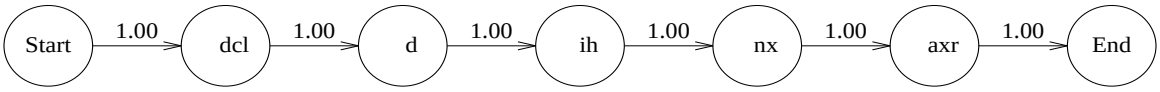


Figure B.11: “dinner”

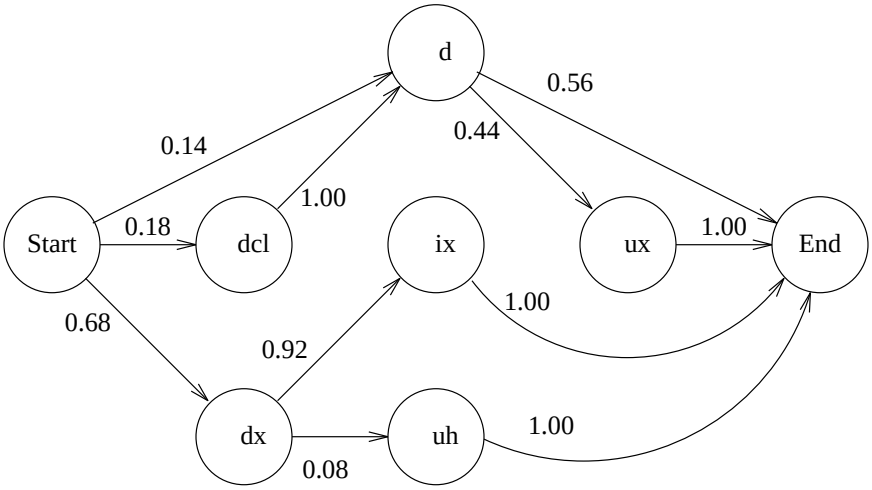


Figure B.12: “do”

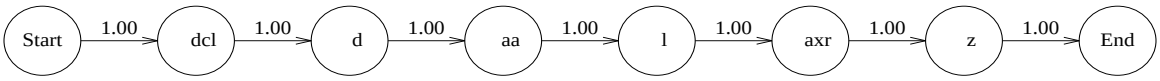


Figure B.13: “dollars”

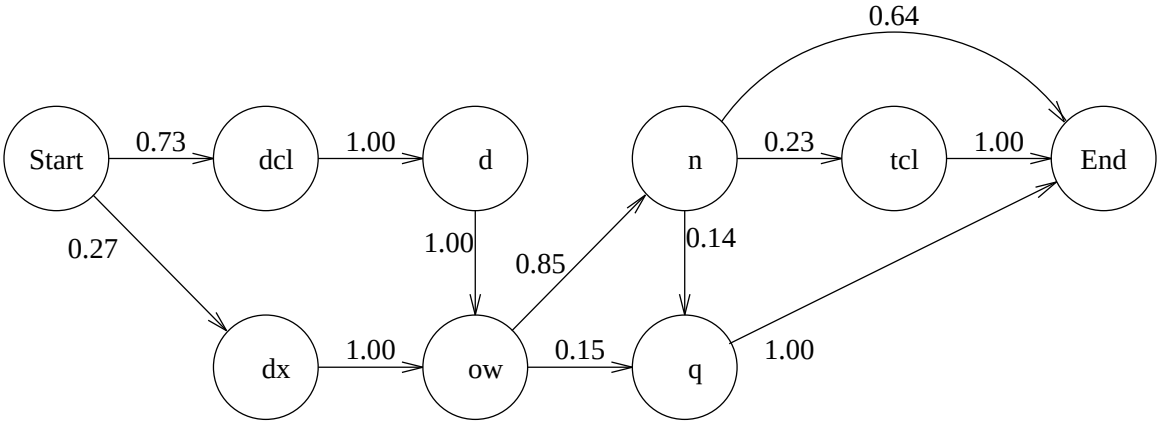


Figure B.14: “don’t”

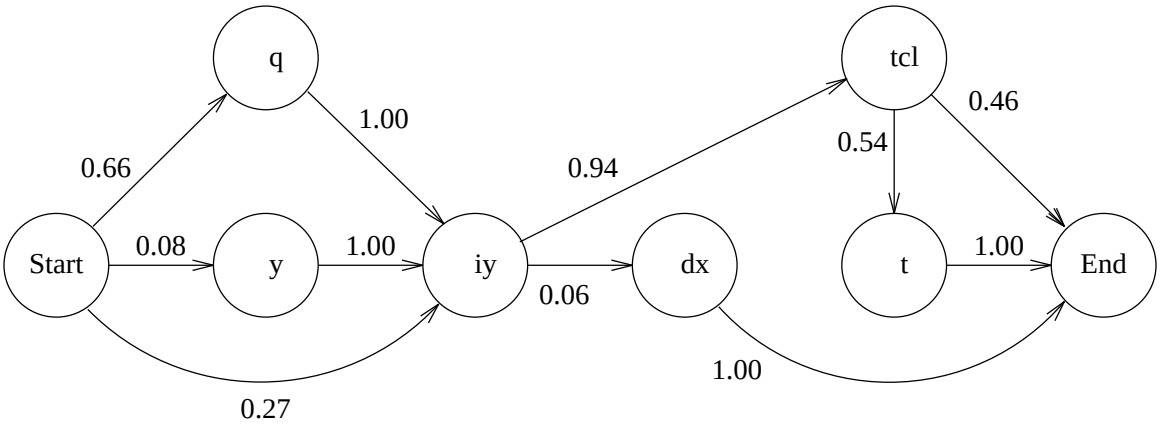


Figure B.15: "eat"

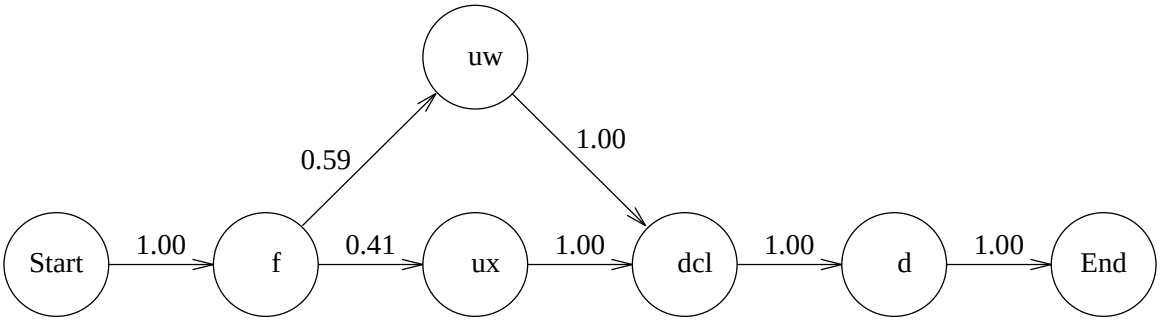


Figure B.16: "food"

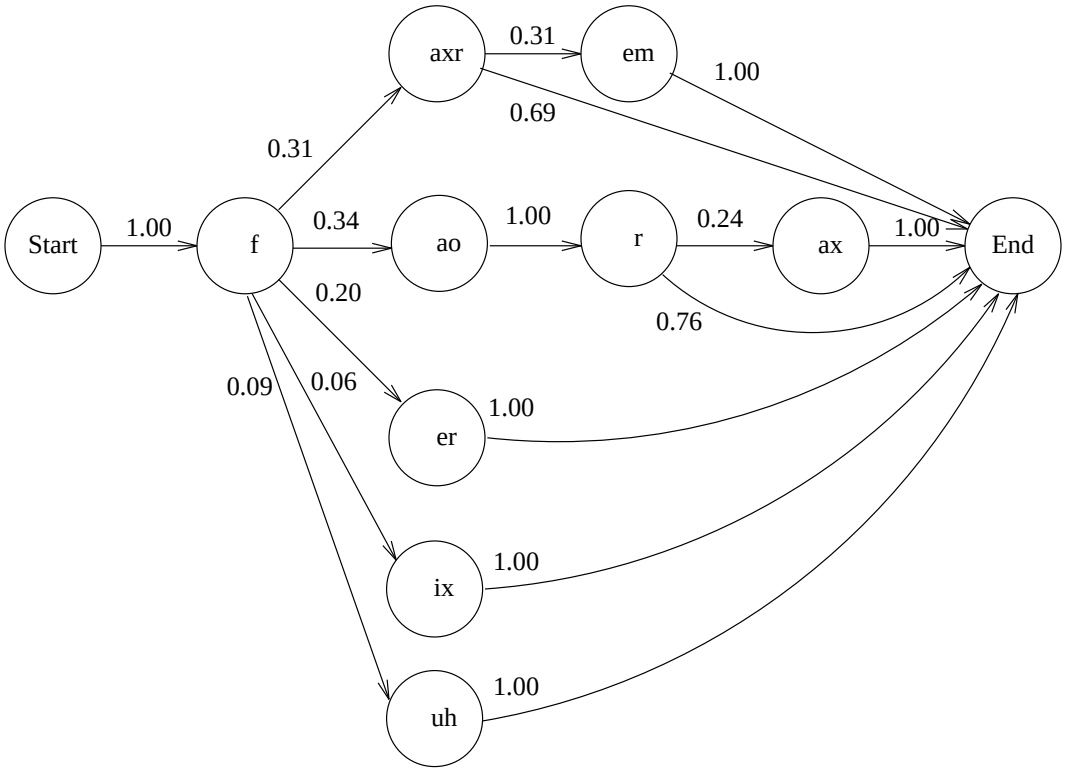


Figure B.17: “for”

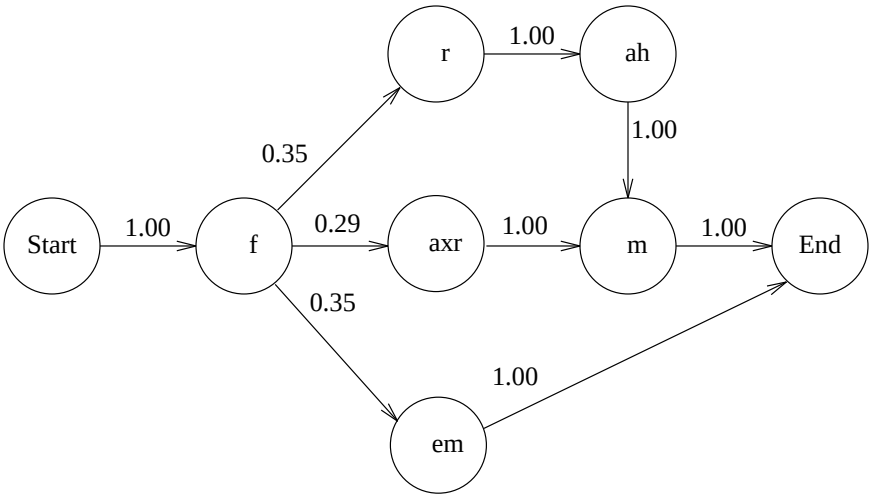


Figure B.18: “from”

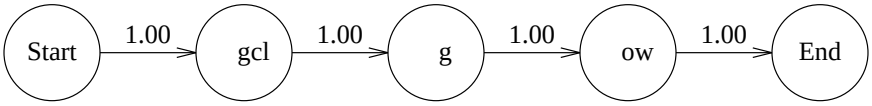


Figure B.19: “go”

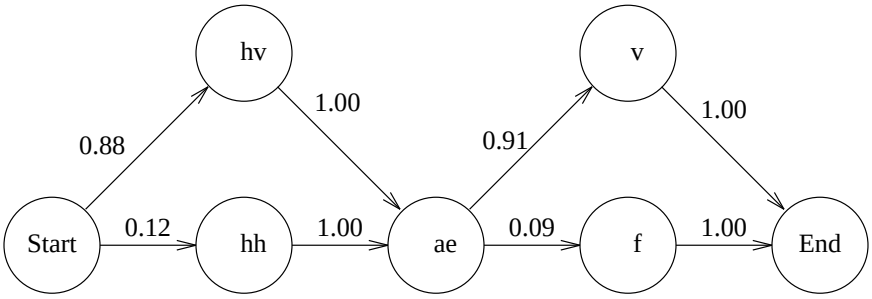


Figure B.20: “have”

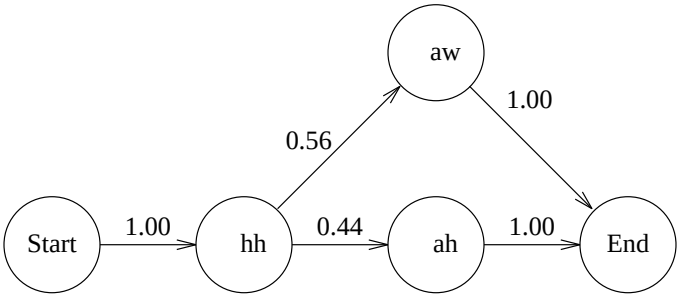


Figure B.21: “how”

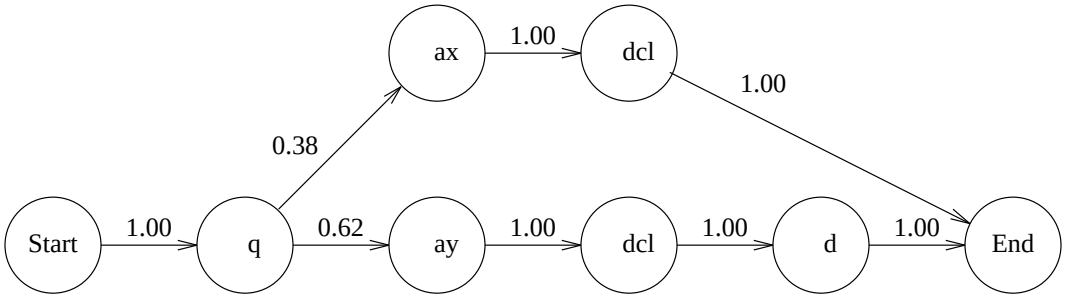


Figure B.22: “i’d”

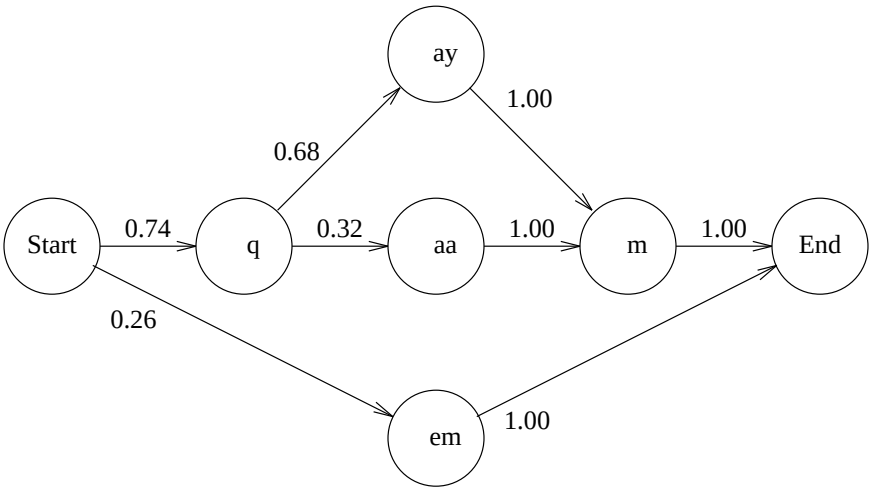


Figure B.23: “i’m”

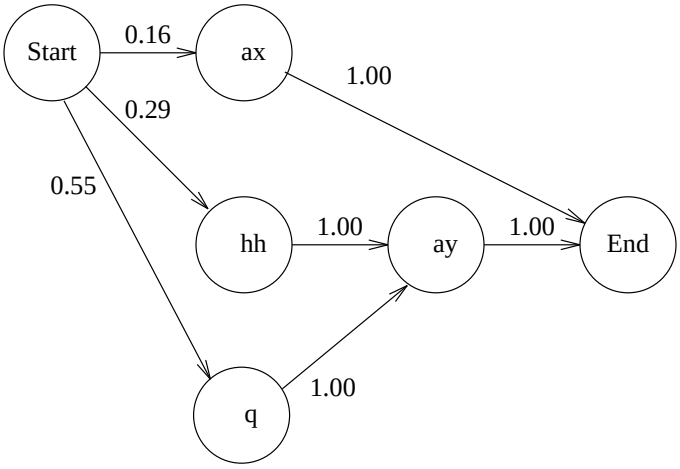


Figure B.24: “i”

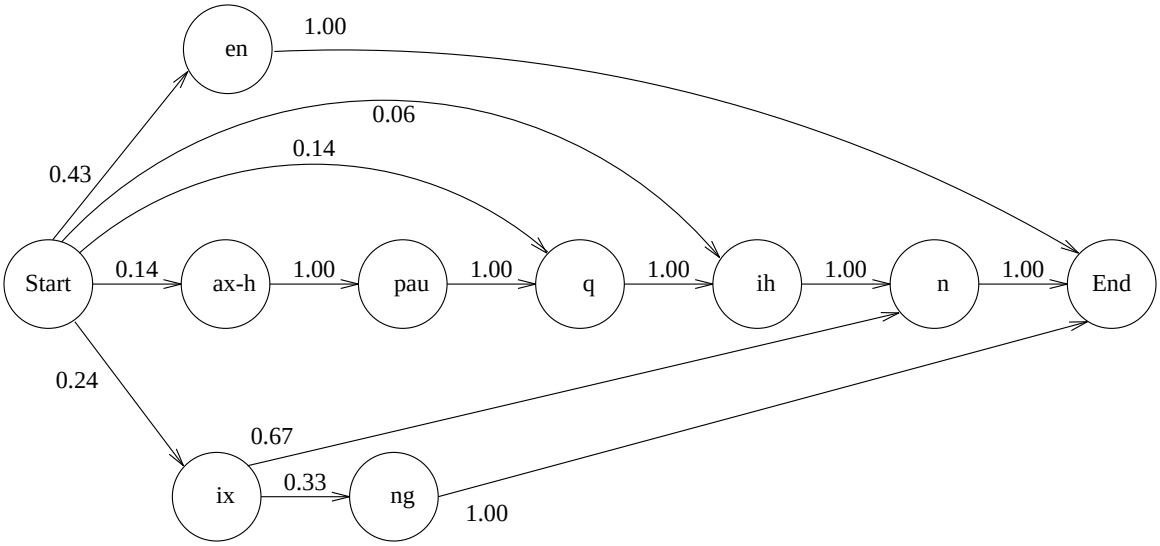


Figure B.25: “in”

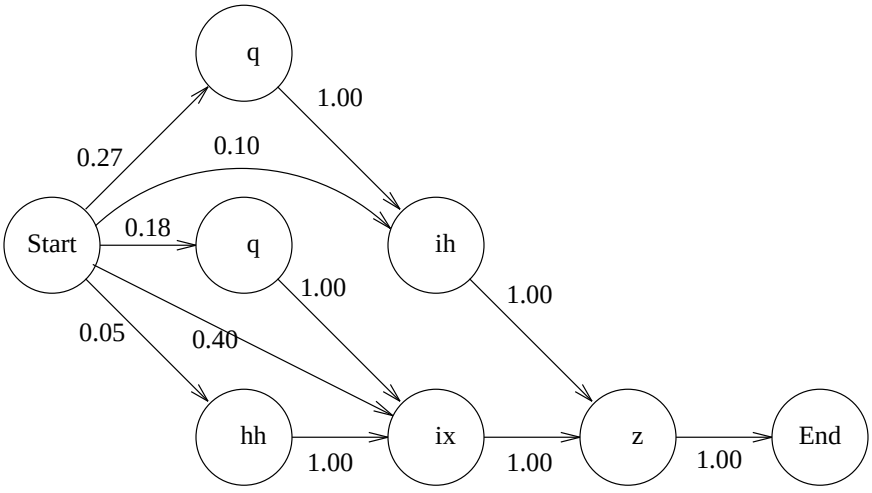


Figure B.26: “is”

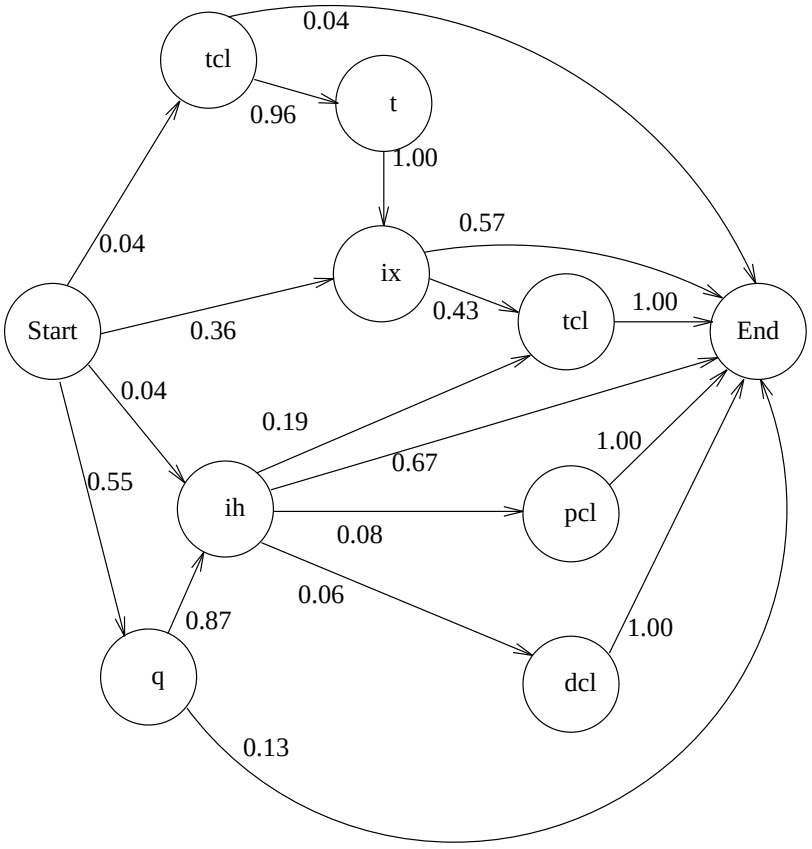


Figure B.27: “it”

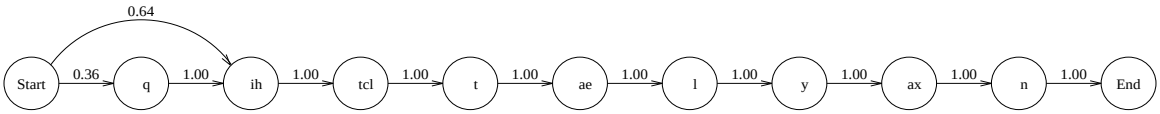


Figure B.28: “italian”

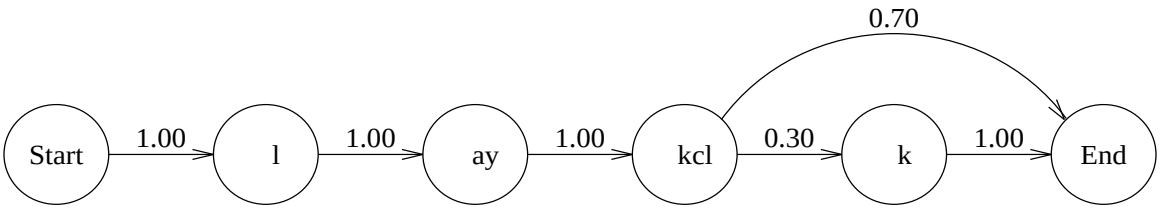


Figure B.29: “like”

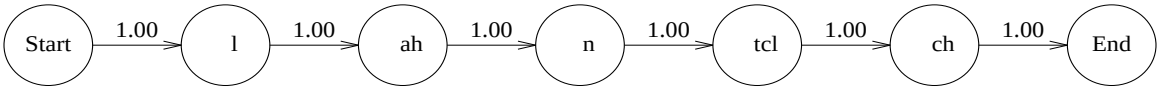


Figure B.30: “lunch”

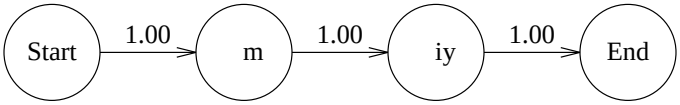


Figure B.31: “me”

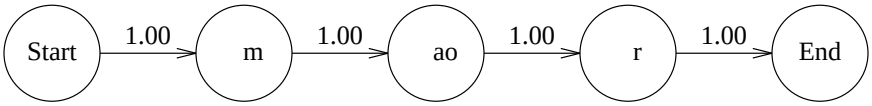


Figure B.32: “more”

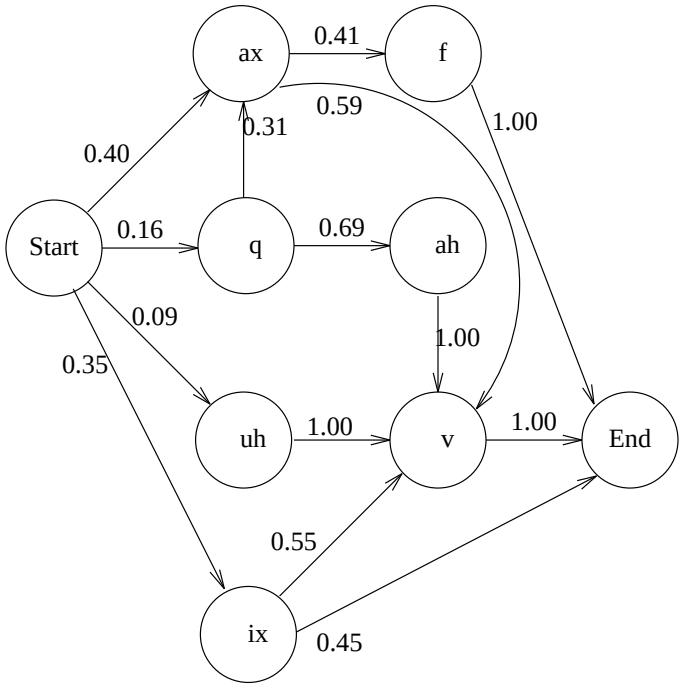


Figure B.33: “of”

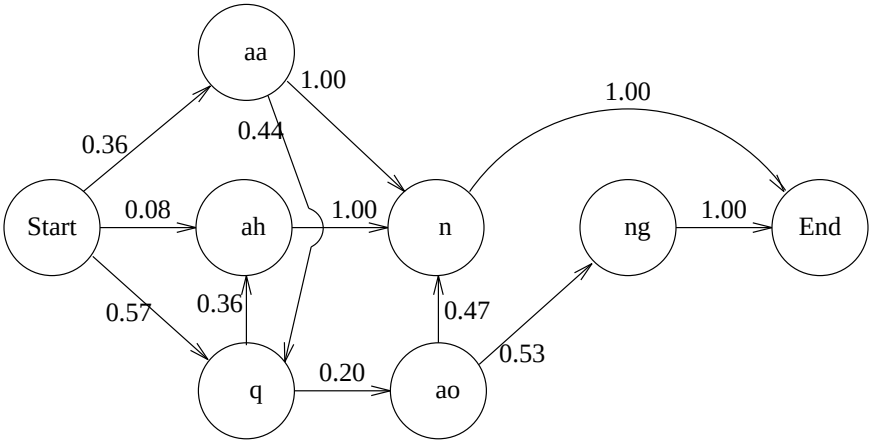


Figure B.34: “on”

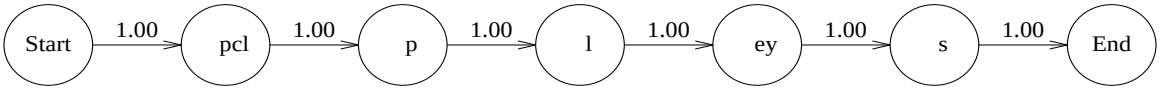


Figure B.35: “place”

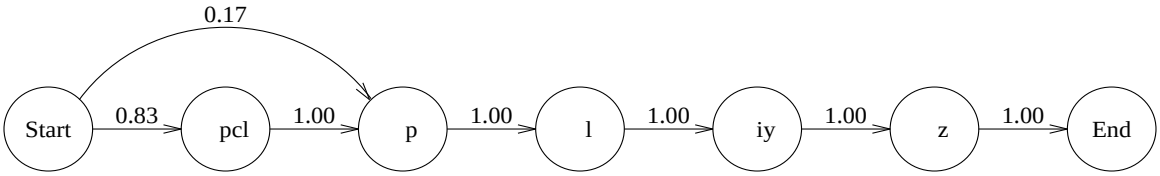


Figure B.36: “please”

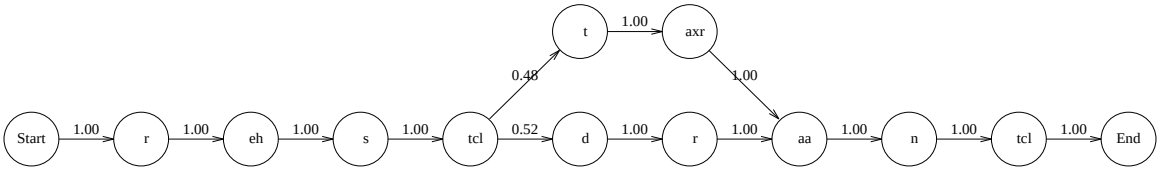


Figure B.37: “restaurant”

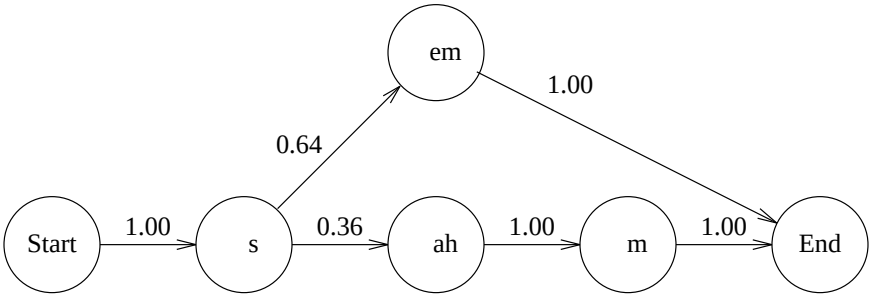


Figure B.38: “some”

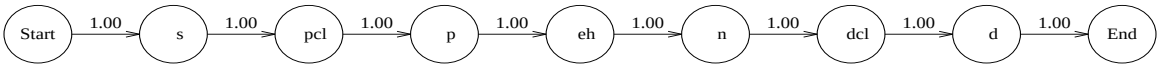


Figure B.39: “spend”

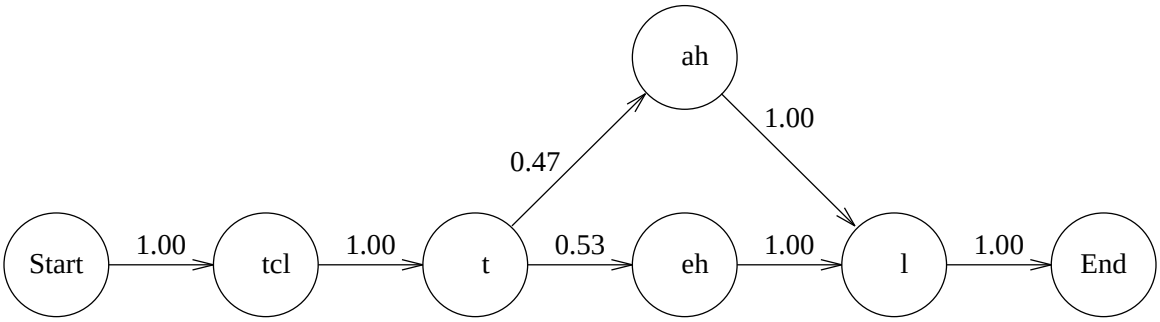


Figure B.40: “tell”

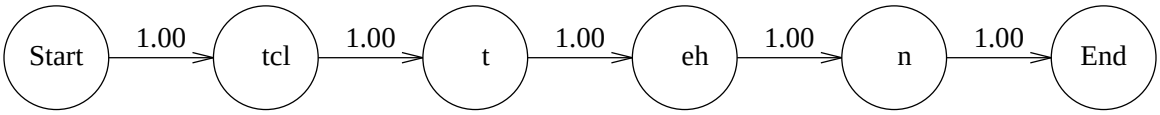


Figure B.41: “ten”

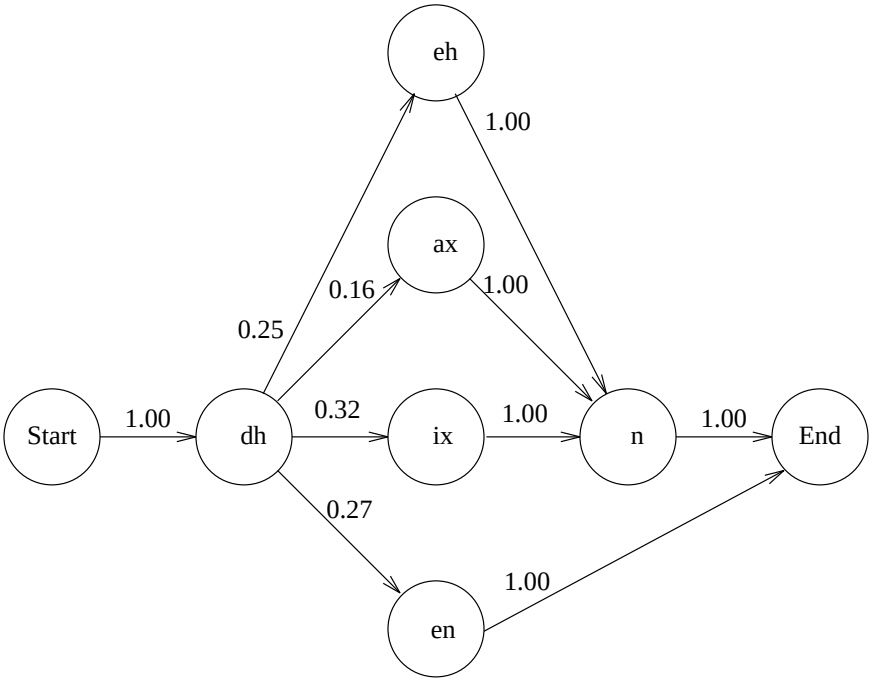


Figure B.42: “than”

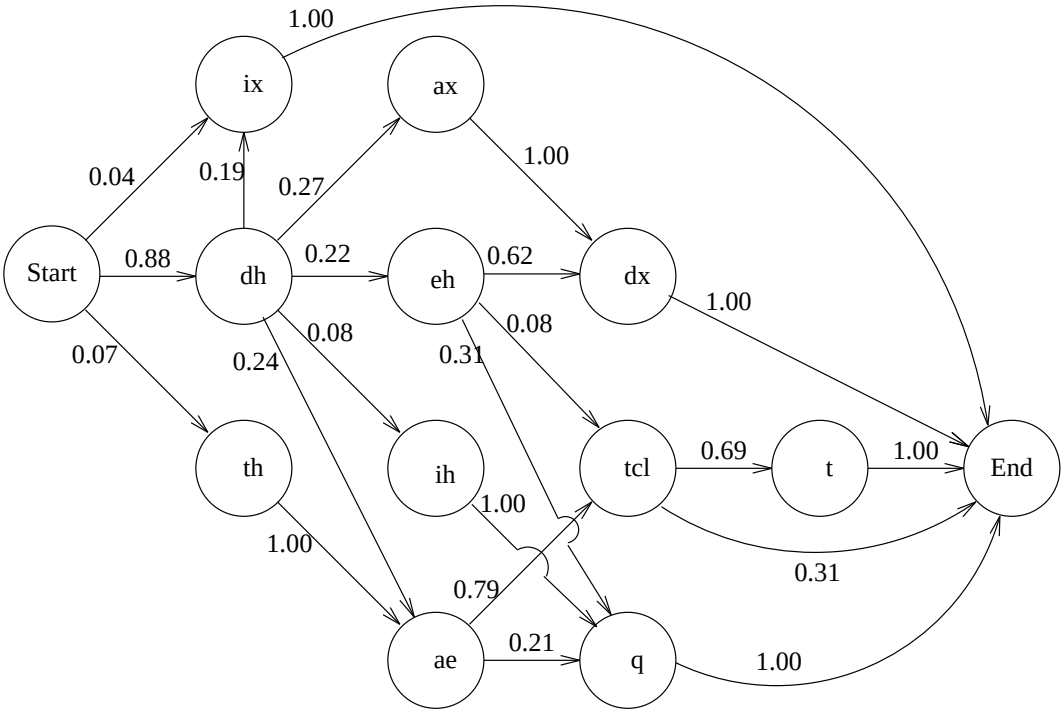


Figure B.43: “that”

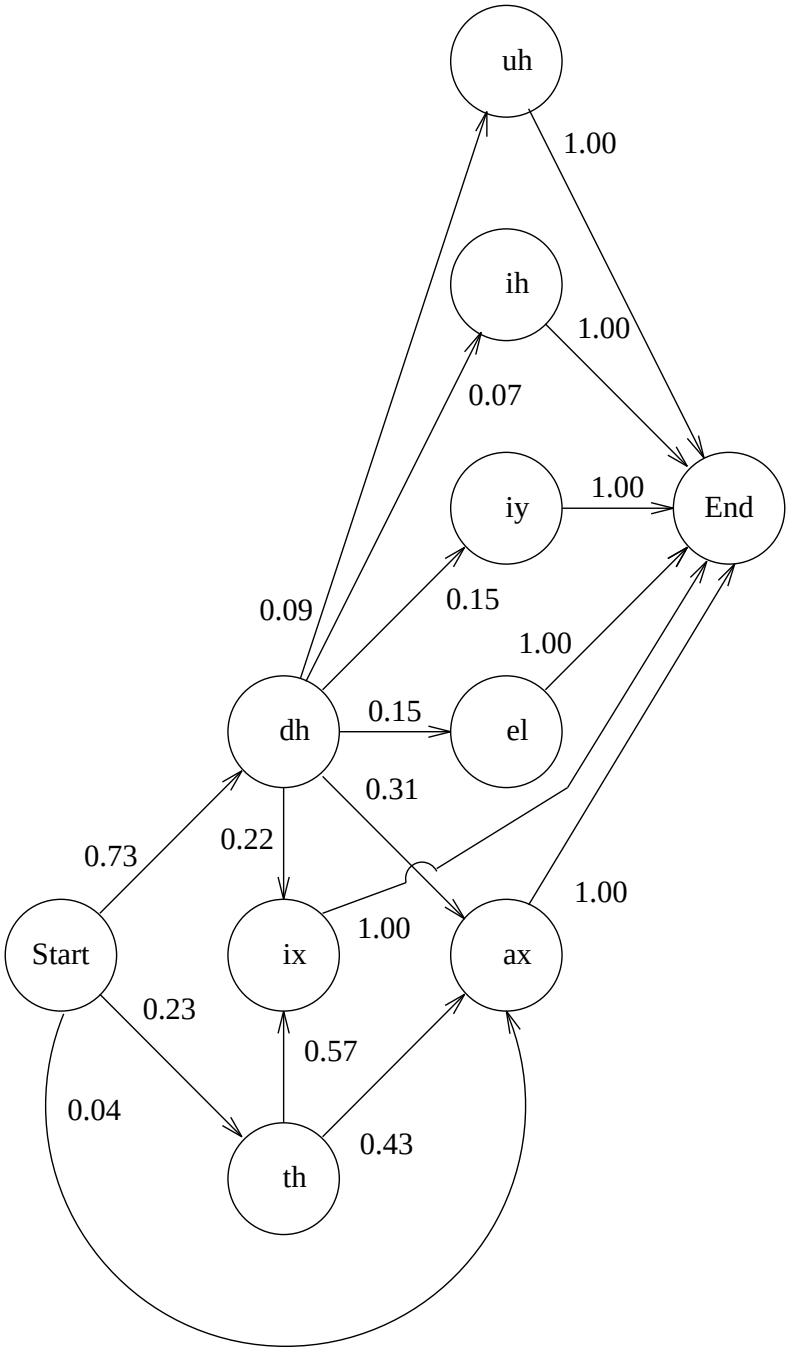


Figure B.44: “the”

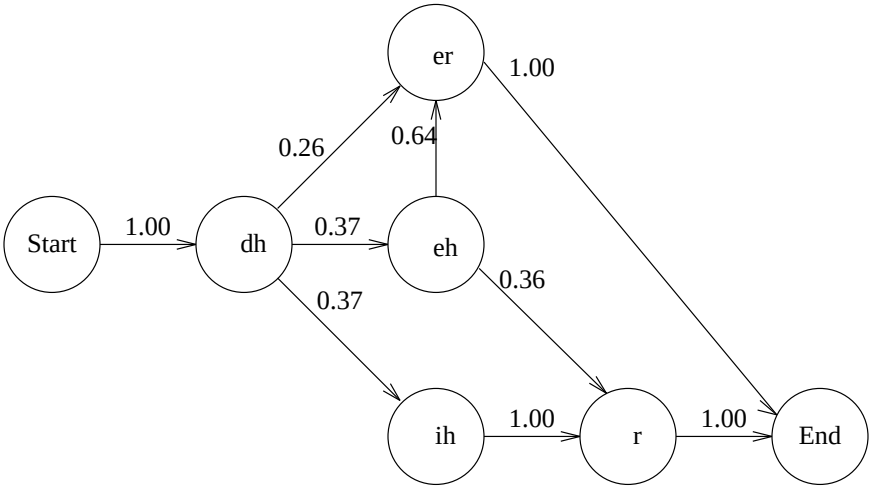


Figure B.45: “there”

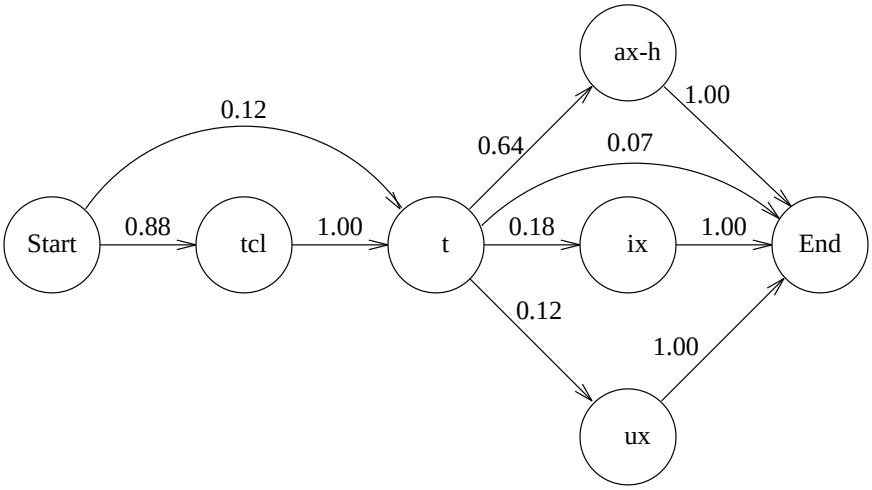


Figure B.46: “to”

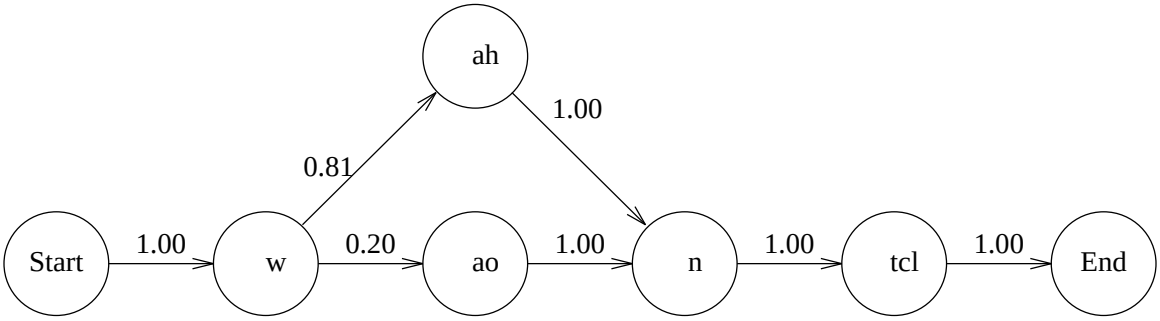


Figure B.47: “want”

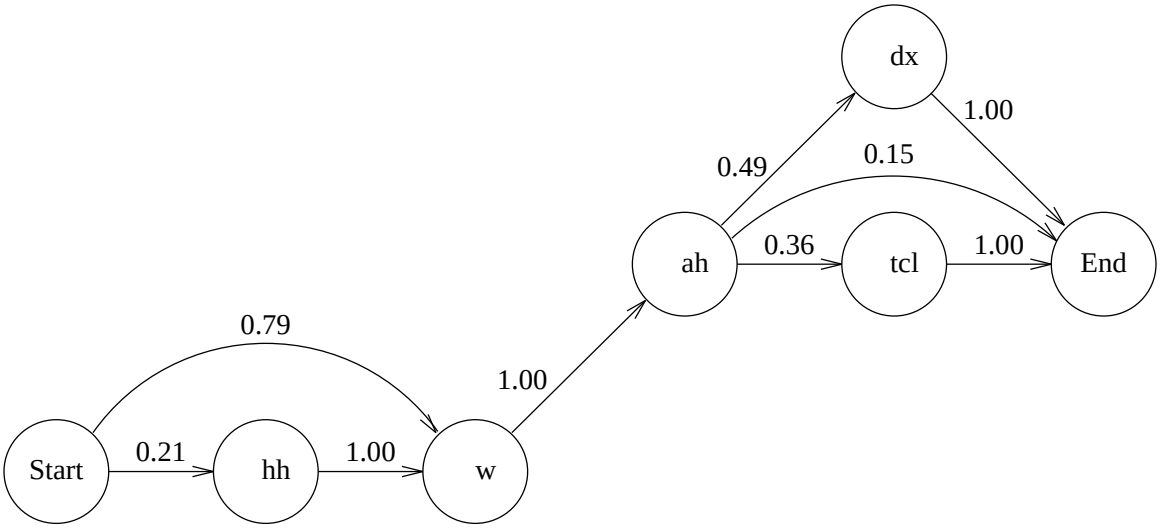


Figure B.48: “what”

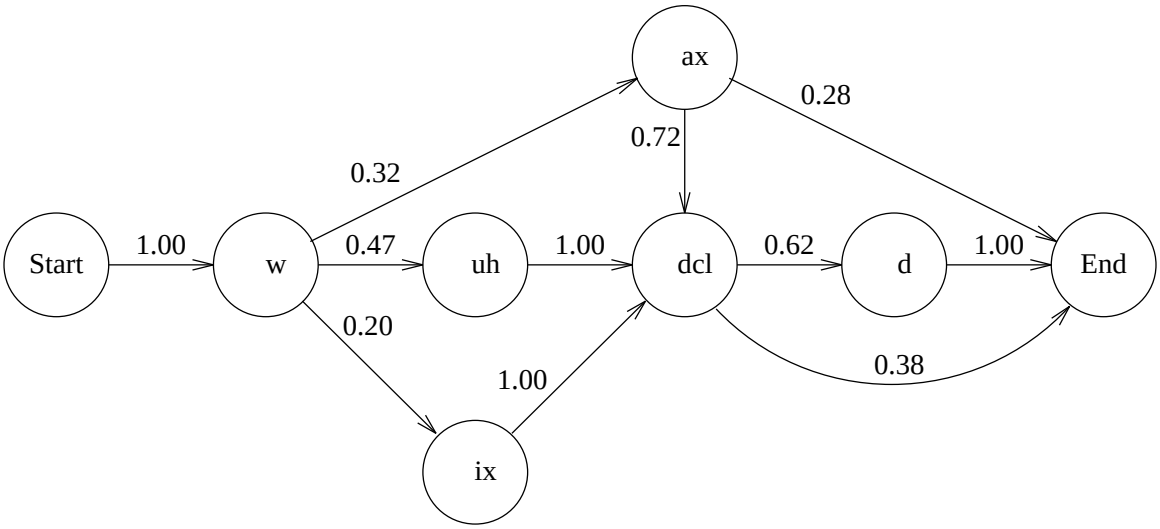


Figure B.49: “would”

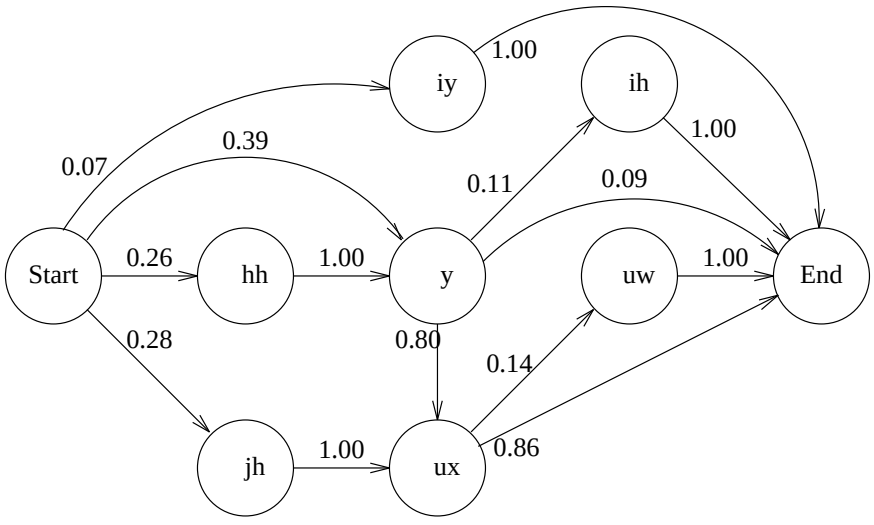


Figure B.50: “you”