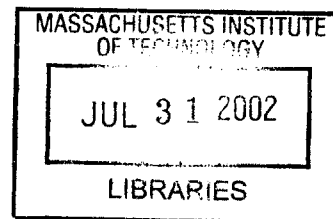


# A Multi-Stage Sound-to-Letter Recognizer

by

Vladislav Y. Gabovich

B.S., Mathematics  
Massachusetts Institute of Technology



Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Masters of Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

©Massachusetts Institute of Technology 2002. All rights reserved.

Author .

.....  
Department of Electrical Engineering and Computer Science  
May 16, 2002

Certified by.

u u

10

.....  
Stephanie Seneff  
Principal Research Scientist  
Thesis Supervisor

Accepted by .....

.....  
Arthur C. Smith  
Chairman, Department Committee on Graduate Students



# A Multi-Stage Sound-to-Letter Recognizer

by

Vladislav Y. Gabovich

Submitted to the Department of Electrical Engineering and Computer Science  
on May 16, 2002, in partial fulfillment of the  
requirements for the degree of  
Masters of Engineering

## Abstract

The task of sound-to-letter recognition of unknown utterances is a difficult and important problem for Natural Language systems. This thesis presents a multi-stage sound-to-letter recognizer that is based on Finite State Transducer (FST) architecture, and relies on components of SUMMIT and ANGIE systems developed in the Spoken Language Systems Group. Specifically, we focus on isolated-word spelling recognition on the Phonebook corpus of single telephone utterances that offer a realistic model of real-life speech. The proposed recognition framework is developed around a new class of sublexical units, called spellnemes, which seamlessly integrate phonemic and spelling information. The advantages of the proposed multi-stage approach include improved performance due to tight linguistic constraint on recognition hypothesis, and promising generalization performance. We achieve a Letter Accuracy Rate of 69.5 percent on out-of-vocabulary utterances, which improves to 89.4 percent for the pure phonetic recognition task.

Thesis Supervisor: Stephanie Seneff  
Title: Principal Research Scientist



# Acknowledgments

First and foremost, I'd like to express my deepest gratitude to Stephanie Seneff, my thesis advisor, for guiding me unerringly to the successful completion of this work. Stephanie's brilliant ideas and motivation kept me on the right track, through all times. Her availability and interest in my work always made me feel like I was part of something significant, and gave me all the support that I could possibly ask for.

Next, I'd like to thank Karen Livescu and Chao Wang for providing extremely helpful assistance along the way, and Grace Chung, for not only her own work that served as the foundation for this thesis, but also for helping out at the most critical stage. Likewise, Karen and Chao willingly sacrificed much of their time to help set up the experiments that were the heart of this thesis – without their help, its completion would be in doubt.

Finally, my office-mates! Brooke, Xialong, and of course, the Dog himself! Our office became my home, and they are the reason why this year turned out a lot more fun than it would otherwise. I'd like to wish them the best – for Brooke to go on to an exciting quest for Ph.D., for Xialong to bring Wall Street to its knees, and for Ed, well, among other things, to successfully hunt down some of those arrogant black-belts.

Of course, my family is the driving force in all my pursuits. To them, I owe everything, and to them I implicitly dedicate all accomplishments. There is one dedication, however, that I'd like to make in silence...

And as a final word, I just hope that this thesis will eventually become a small part of my contribution to science – perhaps this is too much to ask for, but as they say, hope dies last.

Dinamo Kiev eto klass!

Dinamo Kiev eto shkola!

Dinamo Kiev - zvezdnyj chas

Sovetskogo futbola!

This research was supported by DARPA, under contract N66001-99-1-8904 monitored through Naval Command, Control and Ocean Surveillance Center.



# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Background . . . . .	15
1.1.1	The Out-Of-Vocabulary Problem . . . . .	16
1.1.2	Placing The Proposed System in Context . . . . .	16
1.1.3	The Sound-to-Letter Problem and General Speech Recognition	17
1.2	Problem Formulation . . . . .	17
1.2.1	Target Language: English . . . . .	18
1.2.2	Utterance Type: Isolated words . . . . .	18
1.2.3	Corpus: PhoneBook . . . . .	18
1.3	Principal Characteristics of the Sound-to-Letter Recognizer . . . . .	19
1.3.1	Probabilistic Modelling . . . . .	19
1.3.2	Data-driven Learning . . . . .	19
1.4	Methods and Objectives . . . . .	20
1.4.1	Thesis Goals . . . . .	20
1.4.2	Overall Approach . . . . .	20
1.4.3	Thesis Roadmap . . . . .	21
<b>2</b>	<b>Toward a Sound-to-Letter Solution</b>	<b>23</b>
2.1	Recognition Tools: an Overview . . . . .	23
2.1.1	The SUMMIT Recognizer . . . . .	24
2.1.2	The ANGIE framework . . . . .	25
2.2	Combining the Recognizers: Acoustics to Phonemics . . . . .	28
2.3	Initial Proposal: Phone-to-Phoneme-to-Letter . . . . .	29

2.4	A New Approach: Phone-to-Phoneme-to-Spellname . . . . .	30
2.5	Summary . . . . .	32
<b>3</b>	<b>Design of the recognizer</b>	<b>33</b>
3.1	Motivating Spellnames . . . . .	33
3.2	Defining Spellnames . . . . .	34
3.3	Design Tasks for the Spellname-based ANGIE Configuration . . . . .	36
3.4	Letter-to-Sound: A Source for Spellname Mappings . . . . .	37
3.5	Parallel Construction of the Letter-to-Phoneme-to-Spellname map (LPSM)	38
3.6	Multi-Stage ANGIE Configuration for Sound-to-Letter Recognition . . . . .	42
3.7	Low-level Rules Construction . . . . .	43
3.8	High-level Rules Construction . . . . .	44
3.9	Summary . . . . .	44
<b>4</b>	<b>Implementation of the Recognizer</b>	<b>45</b>
4.1	Implementation Overview . . . . .	45
4.2	Recognizer Construction . . . . .	46
4.2.1	Data Preparation: PhoneBook Corpus . . . . .	48
4.2.2	Training, Development and Test Sets . . . . .	48
4.2.3	ANGIE Domain Preparation . . . . .	49
4.2.4	ANGIE Training and FST Construction . . . . .	50
4.2.5	Reusability of Recognizer Components . . . . .	51
4.3	Recognizer Evaluation . . . . .	52
4.3.1	Batch-run Architecture Overview . . . . .	53
4.3.2	Recognizer Construction for Normal and Forced Recognition Experiments . . . . .	53
4.3.3	Testing the Recognition Framework . . . . .	55
4.3.4	Analyzing the Recognition Performance . . . . .	56
4.3.5	Batch-run Implementation . . . . .	57
4.4	Implementation Languages . . . . .	58
4.5	Summary . . . . .	59

<b>5</b>	<b>Recognizer Evaluation</b>	<b>61</b>
5.1	Performance Measures . . . . .	61
5.2	Recognizer Optimization . . . . .	63
5.3	Recognition Experiments . . . . .	65
5.3.1	Overview of the Experiments . . . . .	65
5.3.2	Optimal Configuration Performance . . . . .	66
5.3.3	Generalization Performance . . . . .	66
5.3.4	Hypothesis N-best List Statistics . . . . .	67
5.3.5	Generalization performance of the Phoneme-to-Spellname ANGIE second-stage . . . . .	68
5.3.6	The Effect of the Phoneme-to-Spellname ANGIE second-stage on Recognition Performance . . . . .	71
5.3.7	Forced Phonetic Recognition (FPR) Performance . . . . .	72
5.3.8	Interpreting Forced Phonetic Recognition Results . . . . .	73
5.3.9	Summary . . . . .	75
<b>6</b>	<b>Conclusions and Future work</b>	<b>79</b>
6.1	Assessment of the Multi-Stage Approach . . . . .	79
6.2	Future Improvements . . . . .	80
6.2.1	Reusability of Recognition Modules . . . . .	81
6.2.2	Extensive Phonetic Training . . . . .	81
6.2.3	FST Back-off Paths . . . . .	81
6.3	Integration of the SLR into other Recognition Frameworks . . . . .	82
6.3.1	Online Lexica/Dictionaries for Spelling Verification . . . . .	83
6.3.2	Context-dependent Language and NL Models . . . . .	83
6.3.3	Solicited Word Spelling/Pronunciation . . . . .	83
6.4	Summary . . . . .	85



# List of Figures

2-1	Typical ANGIE parse tree(s) for the word “novel”, showing both phone-to-phoneme and letter-to-phoneme terminal alignments. In bold on the right is the corresponding sublexical layer hierarchy. . . . .	26
2-2	The initial sound-to-letter proposal: modified concatenation of the two common ANGIE frameworks: phone-to-phoneme and inverted letter-to-phoneme. . . . .	30
2-3	The final sound-to-letter proposal: A SUMMIT phone graph composed with two ANGIE-based FSTs – phone-to-phoneme for the modelling of phonological rules, and phoneme-to-spellname in the role of the language model. . . . .	31
3-1	Deducing the spellname-based framework: the top right part shows the constraints which defined the place of the spellname in the sublexical hierarchy; the top left part describes how the set of spellnames was derived from a common ANGIE configuration. . . . .	36
3-2	Example of an ambiguous one-to-many alignment for the phoneme <i>aor+</i> , in the context of the letter “o”. . . . .	39
4-1	The overall sound-to-letter recognition framework . . . . .	46
4-2	Stages of recognizer construction. . . . .	47
4-3	Recognizer training: <code>train-slr.cmd</code> schematics . . . . .	51
4-4	A brief flow-chart of recognition phases. . . . .	53

4-5	The two main recognizer testing configurations: Forced – phoneme-to-spellname recognizer (Top) and Normal/Acoustic – phone-to-spellname recognizer (Bottom). . . . .	54
4-6	Recognizer Evaluation: <code>slr-batch-test.cmd</code> schematics. . . . .	55
5-1	Recognition accuracy (LAR) as a function of component FST weights. The performance of the fully-trained recognizer, evaluated on the first 1000 utterances of the development set. . . . .	64
5-2	Cumulative N-best depth counts for correct hypotheses. Each curve corresponds to a particular training configuration (1st stage training set : 2nd stage training set). A particular point on a curve describes how many correct recognition hypotheses (Y-axis) appeared in the recognizer’s N-best list before or at the given depth (X-axis). For a total of 7500 words, at most 2500 correct hypotheses were found, given the top-performing configuration and a maximum N-best depth of 50. . .	69
5-3	Decomposition of test set performance of a fully-trained (“fulltrain” for both ANGIE stages) recognizer according to forced phonetic recognition subsets. The boxed numbers are the LARs for individual subsets. Decomposition fractions for each subset are shown both as percentages of the parent sets and as percentages of the overall testing set (the later appear in parentheses). . . . .	73
6-1	A roadmap of the integration of the proposed sound-to-letter module into a larger recognition framework. . . . .	82

# List of Tables

2.1	Excerpts from the <i>phonebook-phones</i> ANGIE domain corresponding to the parsing of the word “novel”. . . . .	27
3.1	LPS mappings involving the phoneme <b>aor+</b> . . . . .	40
5.1	Recognition hypothesis/reference alignment for the word “fragmental”. (S)ubstitution, (I)nsertion, and (D)eleation errors are listed below their occurrences, followed by the error rate statistics for the word. . . . .	62
5.2	Generalization performance of a fully-trained recognizer. The LER indicator is the Letter Error Rate, combining substitution, insertion, and deletion error statistics. . . . .	67
5.3	Generalization performance of a recognizer with a complete language model. . . . .	70
5.4	Effect of the second ANGIE phoneme-to-spellneme stage on recognition performance. . . . .	71
5.5	Forced phonetic recognition (FPR) experiments: the performance of the recognizer given correct input phonetics, for the parsed alignments in the development and test sets. . . . .	72
5.6	Normal recognition performance of principal FPR subsets on the test set. . . . .	74



# Chapter 1

## Introduction

### 1.1 Background

Recent work on spoken language systems has yielded a promising array of prototypes in a variety of domains, including weather information [11], flight scheduling [12], and air travel information [7]. These systems propose different models of human-machine interaction, but the recognition of individual utterances remains to be a fundamental problem.

Utterance recognition in a broad sense is the process of mapping a particular instance of an utterance to a singular identifying representation. These instances may take many forms, such as pronunciations, keypad strokes, and/or visual information, among others. The recognition mechanism usually consists of two components – low-level domain-independent acoustic recognition, and high-level domain-specific language modelling. The goal of domain-independent modules is to model patterns common to a language as a whole, whereas a domain-specific approach takes advantage of additional contextual information in order to identify linguistic phenomena common to a particular subset of the language.

This thesis focuses on low-level acoustic recognition, with the overall goal of identifying utterance waveforms. Specifically, we have developed a low-level sound-to-letter recognizer (SLR) that maps waveforms of individual words into an N-best list of possible spellings. The spelling hypotheses can then potentially be used in a second-stage

conjunction with extensive lexicons and/or context-driven language models to identify the utterances. The proposed system is intended as a module in a larger recognition framework, and expands on the work and research tools of the MIT Spoken Language Systems group (SLS). Its primary features include domain-independence, and the ability to handle out-of-vocabulary (OOV) words. Our sound-to-letter recognizer is based on a multi-stage approach, motivated by the desire to achieve maximum linguistic constraint, and thus recognition accuracy, while retaining flexibility and generality. In order to achieve this dual goal, we rely on a new linguistic unit, the *spellname*, that combines phonetic and spelling information in a singular representation.

### 1.1.1 The Out-Of-Vocabulary Problem

It has been discovered that in many domains the occurrence of out-of-vocabulary (OOV) words is significant, and remains substantial as the vocabulary size increases, even for large vocabularies [8]. Mainly, the presence of OOV words is sustained by an influx of nouns and proper nouns, which are too numerous to be enumerated *a priori*. Thus, an essential goal of a recognition system is the ability to “learn” new words, eliminating the manual labor of entering new words together with their complete lexical, syntactic and semantic properties. Another reason for the importance of accurate (OOV) recognition is the fact that unknown words result in error-propagation throughout an utterance, as they frequently disturb a recognizer’s hypotheses for surrounding words.

### 1.1.2 Placing The Proposed System in Context

A long-term vision that we aspire towards is the ability for a speech recognition system to automatically recognize occurrences of new words. In particular, the system should be able to propose a spelling and pronunciation for any new word, and then add it to the existing vocabulary. While this vision is beyond the scope of this thesis, we focus on a specific component that we envision as one of the building blocks in the overall

system. We focus on the technological task of automatically proposing a spelling and pronunciation for words spoken in isolation, without any prior explicit lexical knowledge of the word. This is a challenging task, especially for a language like English, which has highly variable letter-to-sound mappings. A further challenge is that we require the spoken utterances to be telephone-quality speech, due to the fact that most of the dialogue system development at SLS is based on telephone access requirements.

### **1.1.3 The Sound-to-Letter Problem and General Speech Recognition**

It is appropriate to view the sound-to-letter task as a significant, but by no means self-sufficient part of a recognition system. Indeed, domain-independence is a mixed blessing. On the one hand, we hope that a recognizer designed to model sublexical events will “learn” the linguistic structure common to the language as a whole. Thus, it should have excellent capability to recognize unknown words, which can prove very challenging for systems whose knowledge is limited to a particular domain. However, this capability induces a performance penalty brought on by the fact that a low-level-only recognizer does not have access to contextual information that would allow a more efficient pruning of its search space. Therefore, in a well-designed recognizer the sound-to-letter module would be augmented with other resources to help constrain the task, such as large on-line lexica or solicited keypad word entry. We explore some of these possibilities in the last chapter.

## **1.2 Problem Formulation**

Our first goal was to formulate precisely the sound-to-letter problem and to define the exact scope of this thesis. The main choices that we faced were concerned with the recognition language, the type of utterances that the proposed recognizer would handle, and the level of realism in speech quality that we wanted to recreate.

### **1.2.1 Target Language: English**

English is a good example of a highly-irregular language – a language with a great variability in its sound-to-letter mappings. It is also the language with the most readily available data. Therefore it is both a convenient and a challenging example of the sound-to-letter task. Moreover, the majority of the research that provides the foundation for this thesis involved recognition of English utterances. For these reasons, English was the obvious language of choice for our experiments.

### **1.2.2 Utterance Type: Isolated words**

A reasonable initial target for sound-to-letter research is the domain of individual words. As mentioned above, most OOV words turn out to be nouns and proper nouns [8] which can be relatively easily separated from a background of words representing other parts of speech, and recognized individually. Moreover, specialization on individual words allowed our research to be focused on word sublexical modelling, eliminating the need to model word boundaries. In the last chapter, we describe how our recognizer could be used as a post-processor for a system that identifies OOV word boundaries.

### **1.2.3 Corpus: PhoneBook**

One corpus that fits the English language and single-word utterance criteria, as well as the requirement for telephone-quality recording conditions, is the Phonebook corpus of single-word utterances collected over telephone lines [5]. This corpus is rich in both vocabulary and the number of alternate pronunciations for each word. Further, its design allows for easy manipulation and referencing of the data. The fact that the word pronunciations were recorded in a realistic telephone environment lends some credibility to the performance of PhoneBook-based systems on real-life data. The PhoneBook corpus is described in greater detail in Chapter 4 of this thesis.

## 1.3 Principal Characteristics of the Sound-to-Letter Recognizer

In this section we review the general properties of the sound-to-letter task, preparing the foundation for the description of our approach to this problem.

### 1.3.1 Probabilistic Modelling

The fact that the recognizer must propose reasonable spellings for OOV utterances in a highly variable language requires it to probabilistically model sound-to-letter relationships. Indeed, the space of possible English sound-to-letter mappings is extremely large, and there exist no compact collections of “hard” rules that can accurately capture most of the sublexical phenomena in this space. Thus arises the need to approximate exact recognition via a probabilistic approach, in which probabilities or weights are assigned to possible letter-sound alignments on the basis of statistical data. The recognition hypothesis is then produced by an algorithm that attempts to maximize the likelihood of proposed spellings, based on the conditional probabilities of component subsequences.

### 1.3.2 Data-driven Learning

While the space of sublexical sound-to-letter relationships is certainly too large to be described manually, it is also very difficult to construct, *a priori*, a probabilistic rule set that would adequately perform two principal tasks: (i) Generalization of linguistic phenomena to unknown words, and (ii) Attainment of high recognition accuracy on particular utterances. Thus, there arises a need to construct a recognition framework in which the inherent sublexical probabilities are “learned” from a set of utterances with the purpose of capturing linguistic patterns across the entire language. This approach gives birth to the training/development/test paradigm of data decomposition. The training set, typically 70-80 percent of available transcribed utterance data, is used to “train” the recognizer by optimizing its probabilistic rules.

The development set is the test-bed for evaluating the recognition performance on OOV words, and thus - gauging its ability to generalize linguistic patterns. Finally, the test set is reserved for reporting final, unbiased system performance that is free of overfitting (a recognizer may be optimized to perform well just on the development set utterances). The words and their pronunciations are disjoint among the three data sets, meaning that all pronunciations for a given word are part of the same set, and every word appears in exactly one of the three sets.

## 1.4 Methods and Objectives

### 1.4.1 Thesis Goals

The purpose of this thesis is to determine the technical viability of domain-independent sound-to-letter recognition and to develop a workable approach for this task. Specifically, our objective is to build a functioning sound-to-letter recognizer, and thoroughly analyze its performance and construction issues. The proposed system architecture is designed to tackle the dual goal of achieving high recognition accuracy via tightly constraining sublexical mappings, while retaining the flexibility and generality necessary to cover unknown words.

### 1.4.2 Overall Approach

The critical design task of this thesis will be centered around integrating a new sublexical unit, the “spellname”, into a multi-stage recognition framework. The spellname is a unique lexical unit that combines local pronunciation and spelling information, thus providing a complete description of an utterance. Our general approach will be to process input utterance acoustics through a series of transducers, obtaining a string of the most likely spellnames, from which the spelling could be readily extracted. We divide the overall sound-to-letter recognition task into three stages, acoustic-to-phonetic, phonetic-to-phonemic, and phonemic-to-spellnemic. All three stages are based on existing SLS tools: the SUMMIT recognizer for the first acoustic-to-phonetic

stage, and the ANGIE probabilistic parser for the remaining two stages. Henceforth, the phone-to-phoneme and phoneme-to-spellneme stages will also be referred to as the first and the second ANGIE stages, or as the phonological rules and language model stages (for reasons that will be made apparent in Chapter 2).

### **1.4.3 Thesis Roadmap**

The current chapter has briefly presented the main characteristics of the sound-to-letter problem, defining the goals of this thesis in the context of a particular set of design constraints. Chapter 2 presents the framework of main recognition components that form the basis of our system, and outlines the main features of the selected approach. Chapter 3 deals with the most essential design issues in the recognizer construction, and reveals the role of an important new sublexical unit, the spellneme, in the overall framework. Chapter 4 describes in detail the architecture of the resulting system, the tools used to glue the complete recognizer together, and various integration issues. Chapter 5 presents the principal questions that our recognizer was built to answer, and the detailed experiments that produced the answers to these questions. Finally, in Chapter 6 we place our work in perspective, consider its logical extensions, and set the stage for future work.



# Chapter 2

## Toward a Sound-to-Letter Solution

Our approach to the sound-to-letter task relies heavily on certain core components developed at MIT's Spoken Language Systems (SLS) Group, and therefore we begin with a discussion of some of these systems and the motivation behind them. Then, we outline one possible approach to the sound-to-letter problem, comment on its viability, and use the insights thus obtained to propose a multi-stage sound-to-letter recognition scheme based on a new sublexical unit, the *spellneme*.

### 2.1 Recognition Tools: an Overview

In the context of the sound-to-letter task, past SLS research had oriented our efforts in the direction of constructing an FST-based recognition framework. This framework would support hierarchical sublexical structure, synthesizing both phonetic and spelling information, and enable the realization of the recognizer in the form of a finite state transducer. Two SLS systems are particularly suited for such tasks - the ANGIE probabilistic framework for sublexical modelling, and the SUMMIT acoustic recognizer for phonetic transduction of input acoustics.

Finite-state transducers (FST) are a relatively new approach to speech recognition tasks. The elegance of FST representation stems from a solid mathematical foundation [6]. Fundamentally, FSTs are graphical models that describe state transition possibilities and the accompanying transition probabilities, mapping/transducing a

sequence of input units into a sequence of output units. Their advantage lies in the simplicity of construction, compactness of representation, modularity (the ability to compose multiple FSTs into multi-stage mappings), and a rich set of fast operations congruent with speech recognition tasks (simulation, hypothesis selection and pruning, optimization). One of the most significant benefits of FST-based recognition is the remarkable ease of integration of different models into a seamless recognition framework [10]. We make full use of this trait in our multi-stage sound-to-letter proposal, and explore some interesting integration possibilities in the final chapter.

### 2.1.1 The SUMMIT Recognizer

A critical tool that provides the foundation for much current SLS research is the SUMMIT speech recognition system [4]. This FST-based recognizer can be used to transcribe acoustic signals into phonemic sequences and subsequently into words and sentences. SUMMIT is a segment-based recognizer capable of utilizing context-dependent diphone boundary models. In the current instantiations for English applications, it uses a total of 68 phonetic units and 631 diphones. In order to calculate acoustic features, the signal is split into 5 msec frames, and 8 different averages of 14 Mel-scale cepstral coefficients are computed from 150 msec windows surrounding each of the frame boundaries. A diagonal Gaussian mixture, with up to 50 kernels, is created for each of the resulting diphone boundary models. Hand-written phonological rules are used to expand a pronunciation network based on the phonetic baseform. The system typically uses a bigram in a forward Viterbi search, and a trigram in a subsequent backward search. In its standard form, the overall SUMMIT recognizer can be represented as the composition of several FSTs:

$$R = P \circ L \circ G,$$

where  $P$  is a phonetic graph weighed with acoustic scores,  $L$  is the lexicon, and  $G$  is the language model. As we will see, SUMMIT can be used as the ideal low-level complement for our higher-level systems (ANGIE), parsing acoustics into a form that can be used by other FSTs for further sublexical analysis. In our work, we utilize

only the low-level part of the typical SUMMIT configuration – a phone graph based only on acoustic and syllable-level information. In the recognition framework that we propose, the SUMMIT phonological rules and language model are replaced by FST modules derived from ANGIE – another principal SLS tool, which we describe next. However, the phonetic sequences that are used to train the phonological component of the ANGIE FST are obtained via a forced alignment step that makes use of the standard phonological rule mechanism in SUMMIT.

### 2.1.2 The ANGIE framework

ANGIE [9] is a powerful platform for the representation of various sublexical linguistic phenomena in a unified framework. It is essentially based on a hierarchical tree structure whose layers capture sublexical phenomena at different levels - morphology, syllabification, phonemes, and phones/letters. Normally, the inter-layer grammars are constructed manually. Depending on recognition requirements, ANGIE can be configured in various ways. Figure 2-1 presents a typical ANGIE parse tree(s) for the word “novel”, showing both a phonetic and an orthographic terminal level alignment, corresponding to the two common ANGIE configurations: letter-to-phoneme and phone-to-phoneme. In this example, a letter/phone sequence feeds the tree terminals, with phonemes as the pre-terminal units.

It is helpful to briefly discuss some key ANGIE terminology that defines the relationships between adjacent layers in the sublexical hierarchy. The following discussion is illustrated in Table 2.1, which details some of the inter-layer mappings for the word “novel”, whose parse trees appear in Figure 2-1.

#### Lexicons

The *lexicons* are mappings with unique domains and ranges, defining dictionaries of units higher up in the hierarchy in terms of the units below them. The two main ANGIE lexicons are the word-to-morph and morph-to-phoneme lexicons. Typically, for each word, there corresponds one and only one morph sequence; and conversely,

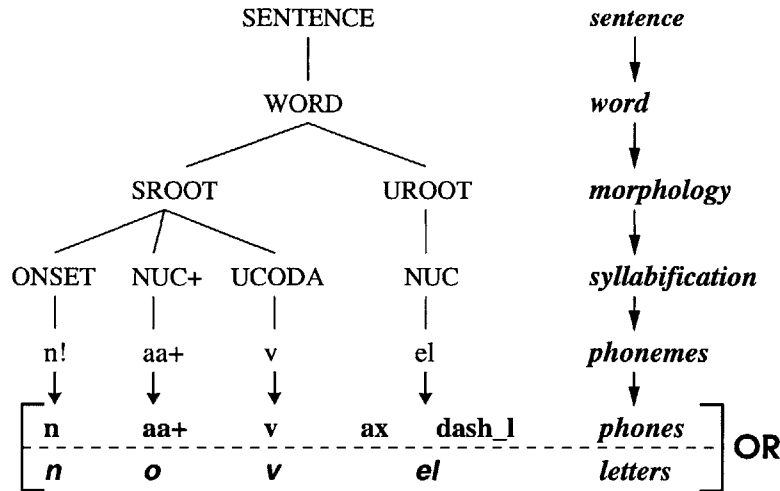


Figure 2-1: Typical ANGIE parse tree(s) for the word “novel”, showing both phone-to-phoneme and letter-to-phoneme terminal alignments. In bold on the right is the corresponding sublexical layer hierarchy.

a particular morph sequence maps to a unique word. The morph-to-phoneme lexicon follows the same one-to-one pattern.

### Grammars

Other mappings require greater flexibility, for units which have many possible realizations in the underlying layer. ANGIE *grammars* are expressed in terms of the high-level rules, which describe possible word decompositions into pre-terminals, and the low-level-rules, which define the potential realizations of pre-terminal units in terms of the terminals. For example, a particular phoneme may be realized by one of several letter sequences, as shown in Table 2.1.

### Alignments

Finally, word-to-(terminal) *alignments* constitute the training data in the ANGIE framework. These alignments constitute the data for the training of the ANGIE recognizer. An alignment represents a known representation of a word in terms of the chosen terminal units. Given an alignment, ANGIE builds up a hypothesis for the most likely corresponding sublexical structure, constrained by its various inter-layer rules. The probabilistic inference is performed in terms of ANGIE tree “columns” (a

<b>Lexicons</b>	
Word to morphs	Morph to phonemes
novel : <i>nOv+ el</i>	nOv+ : <i>n! aa+ v</i>
	el : <i>el</i>
<b>Grammars</b>	
Word to phonemes (High-level-rules)	Phoneme to phones (Low-level-rules)
word [pre] sroot [uroot] ...	n! ( <i>n nx</i> )
onset (p! t! .. n! ..)	aa+ ( <i>aa aa-f ao ao-f ..</i> )
<b>Alignment</b>	
Word to phones	
novel : <i>n aa+ v ax dash-l</i>	
novel : <i>n aa+ v el</i>	

Table 2.1: Excerpts from the *phonebook-phones* ANGIE domain corresponding to the parsing of the word “novel”.

sequence of units on a path from the top to the terminal layer), whose likelihoods are ultimately represented in an FST in terms of column-column transition probabilities. Thus, the “best” hypothesis corresponds to a sequence of the most likely “columns”. The training ANGIE over a set of alignments is an iterative process: counts are maintained on the observed pairs of adjacent columns, and then used to determine the best column sequences for subsequent alignments. More information on the ANGIE framework can be found elsewhere [9].

## Domain

The sum total of grammars, lexicons, and alignment files for a particular ANGIE configuration forms the ANGIE *domain*. Typically, the locations of the various components are specified in a domain file that controls the operation of ANGIE training and parsing algorithms. Table 2.1 shows some excerpts from files constituting the phone-to-phoneme ANGIE domain.

The column-bigram probabilities obtained from training ANGIE on a set of alignments can be used to hypothesize the full sublexical structure for input sequences of terminal units. In the final step the best parse tree is usually transformed into a compact FST that maps the terminal input sequence into a sequence of corresponding pre-terminals. For example, in the letter-to-sound ANGIE configuration (a conceptual inverse of our sound-to-letter problem) input strings of letter terminals are mapped into sequences of phonemes, providing a phonemic representation of utterance spellings.

In this thesis, we will mostly be concerned with modifying the two bottom-most layers in the ANGIE hierarchy, henceforth referred to as the terminal and pre-terminal layers. The upper layers model word morphology and syllabification – units which describe general sublexical structure, and do not carry enough word-specific information to encode word pronunciation or spelling.

## **2.2 Combining the Recognizers: Acoustics to Phonemics**

As mentioned above, SUMMIT and ANGIE can be combined to form a complete recognizer that can be trained up to parse diphone acoustics into a complete sublexical structure. The speech recognition task is an example of one such combination. As one may recall from section 2.1.2, in the speech recognition context ANGIE can be trained to align phoneme pre-terminals to phone terminal units. In this case, the column-bigram FST constructed from the trained ANGIE grammar transduces phonetic sequences into strings of phonemes. FST composition enables us to integrate this phone-to-phoneme transducer with an acoustic recognizer by composing the SUMMIT diphone-to-phone and ANGIE phone-to-phoneme FSTs. This procedure yields a complete recognizer that can parse utterance acoustics into strings of phonemes.

The power and compatibility of the SUMMIT and ANGIE models motivated

us to seek a solution for the sound-to-letter task that would integrate instances of both systems. SUMMIT was to be used as a bridge between the utterance acoustics and phonetic sequences. ANGIE would be called upon to transduce the resulting SUMMIT phonetic transcriptions of words into spelling hypotheses, according to its sublexical probabilistic grammars. Our major challenge was to modify the ANGIE framework in a way that would enable the extraction of spelling information from the hypothesized sublexical hierarchy of parsed words.

## 2.3 Initial Proposal: Phone-to-Phoneme-to-Letter

The most obvious approach to the task of sound-to-letter modelling in the context of ANGIE came from considering the two existing ANGIE configurations – phone-to-phoneme and letter-to-phoneme. Specifically, we began by considering the possibility of reversing the letter-to-phoneme mapping in order to obtain a sound-to-letter transducer. A key property of FSTs is that they can easily be inverted, reversing the input-output roles. Thus, one way to generate spelling hypotheses is to reverse the letter-to-phoneme FST, obtaining a phoneme-to-letter transduction. The resulting FST can then be composed with the usual phone-to-phoneme transducer to obtain an overall phone-to-letter transduction (Figure 2-2).

However, this approach is beset with several problems. Most notably, letters are not very specific sublexical units – they allow many possible phonemic representations, and thus are not very constraining on the space of ANGIE column-bigram hypotheses. As a rule, the ambiguity resulting from such lack of constraint leads to a large number of low-probability hypotheses, resulting in a decrease in recognition performance. In our realization of the sound-to-letter recognizer, one of the principal goals was to create a framework with maximum linguistic constraint, in order to reduce the search space and prune away as many unlikely hypotheses as possible. Based on the lack of constraint in low-level letter-to-phoneme mapping, we expected the phone-phoneme-letter approach to the sound-to-letter task to fare poorly in practice. Therefore, we were motivated to seek an alternative route.

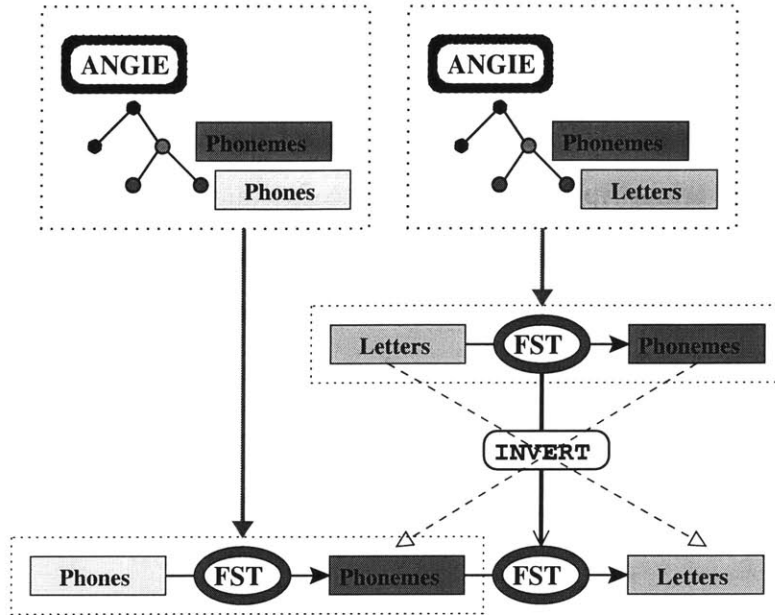


Figure 2-2: The initial sound-to-letter proposal: modified concatenation of the two common ANGIE frameworks: phone-to-phoneme and inverted letter-to-phoneme.

Despite the fact that this initial model seemed inherently flawed, it helped us identify a potentially better approach to sound-to-letter recognition. Specifically, we began the search for a new sublexical unit that would contain spelling information while retaining the properties of phonemes in the ANGIE hierarchy. Thus, a sequence of these new units would be able to yield a spelling for an utterance, and fulfill a useful sublexical purpose, helping constrain the recognition hypotheses. Naturally, our attention turned to the *spellneme*, a sublexical unit first introduced by Grace Chung [3]. This unit performs exactly the desired function of synthesizing phonemic and spelling information. Moreover, the spellneme provides the benefit of additional linguistic constraint without sacrificing the generalization capabilities of the recognizer. Therefore, we turned to the question of incorporating the spellneme into the overall recognition scheme.

## 2.4 A New Approach: Phone-to-Phoneme-to-Spellneme

The primary challenge of incorporating spellnemes into the ANGIE framework lay in exposing their true sublexical function, which translated into finding their place in the

ANGIE sublexical hierarchy. In general, recognition requires the fulfillment of two essential but often conflicting tasks - generalization ability and recognition accuracy. For a static, unchanging domain, it is often possible to build very accurate recognizers; however, the OOV problem demands recognizers with a very large coverage - and as a rule some performance has to be sacrificed to obtain the necessary flexibility. In our task, the decision of how to model the role of spellnemes in sublexical decomposition had a critical effect on the performance and coverage properties of the resulting system.

Careful analysis of the characteristics of our sound-to-letter task, the capabilities of available recognition tools, and the issues uncovered in our initial approach, motivated us to propose a multi-stage recognizer framework. As before, we decided to rely on SUMMIT to supply a mapping from input acoustics to phonetic sequences. However, the final phase of spelling recognition was decomposed into two ANGIE stages - an initial phone-to-phoneme transduction followed by a phoneme-to-spellneme stage, as diagrammed in Figure 2-3.

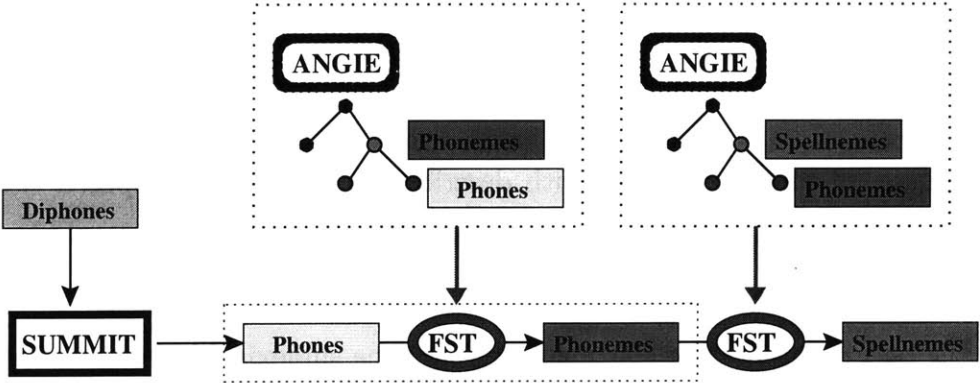


Figure 2-3: The final sound-to-letter proposal: A SUMMIT phone graph composed with two ANGIE-based FSTs – phone-to-phoneme for the modelling of phonological rules, and phoneme-to-spellneme in the role of the language model.

Several important factors drove this design decision. First, as we show in the next chapter, we identified the ANGIE pre-terminal layer as the source of spelling information. This necessitated the creation of an end-of-the-line transducer that would align spellnemes in the pre-terminal layer with one of the existing unit types as terminals. A straight phone-to-spellneme approach was dropped in favor of a more

constraining multi-stage system. We identified the phoneme as the sublexical unit that most naturally fit the role of a bridge between the phone sequences and the spellnemes. Therefore, the phonemic layer became the glue in the composition of the two FSTs, acting in its normal pre-terminal role in the first stage phone-to-phoneme FST, and as a terminal unit in the second phoneme-to-spellneme stage. This choice offered several significant advantages:

1. The ability to use a known, well-defined phone-to-phoneme ANGIE configuration as the first stage.
2. A phoneme-to-spellneme stage readily constructable from a sublexical mapping that implicitly defined the low-level grammar.
3. Improved performance obtained from applying “double” linguistic constraint via the two recognition passes.
4. Clarity of implementation and testing - the consequence of a clean, modular design.

## 2.5 Summary

The remainder of this thesis deals with describing in detail the proposed two-stage ANGIE-spellneme recognizer. The next chapter describes the process of designing a recognizer framework around the spellneme units and existing recognition tools, as well as the task of integrating the spellneme into the sublexical hierarchy. Chapter 4 presents the technical issues of recognizer implementation and testing, and Chapter 5 completes our account of the multi-stage sound-to-letter framework with an analysis of obtained recognition performance.

# Chapter 3

## Design of the recognizer

The approach outlined in Chapter 2 calls for the integration of SUMMIT and ANGIE systems into a multi-stage sound-to-letter recognition framework. The role of SUMMIT will be acoustic-to-phone transduction, while ANGIE will be employed to handle higher-level sublexical phenomena – phonemics, syllabification and morphology. There are multiple advantages in decomposing recognition into separate stages. Most importantly, we intended to use multiple recognition passes in order to impose a sequence of tight linguistic constraints on the recognition hypotheses, thereby preventing the system from running into overgeneralization problems detrimental to recognizer performance. This approach would also be more conceptually clear, enable us to build on existing recognizer configurations, and allow for more precise optimization of system parameters, whose effect could be analyzed separately by keeping independent variables constant. These reasons were particularly enticing in light of the fact that we were going to rely heavily on a new linguistic unit, the spellname, whose properties and function were not yet fully understood.

### 3.1 Motivating Spellnames

We identified the following main design issues for the proposed recognizer:

1. Modelling spelling: loading graphemic (spelling) information into the ANGIE sublexical framework.

2. Maximizing linguistic constraint: introducing tight low-level constraints to improve recognition performance.
3. Preserving generality: keeping the recognition framework domain-independent, capable of parsing unobserved phonetic sequences.

There is a principal tradeoff between tightening constraints, which often increases domain-specificity, and attaining greater flexibility, which usually implies loss of contextual information with a corresponding negative effect on performance. One way to enhance linguistic constraint with low-level domain-independent knowledge is by augmenting the set of linguistic units with grapheme information. In this case, loading the sublexical framework with richer contextual spelling knowledge leads to more precise models without sacrificing domain-independence. At the same time, the dual goal of obtaining spelling information is satisfied. The task of integrating the augmented units into the ANGIE framework in such a way as to model both phonological and sound-to-letter rules provides the primary motivation for the development of the “spellname” sublexical unit.

## 3.2 Defining Spellnames

The “spellname” unit <sup>1</sup>, also referred to as a “grapheme” or a “letter-phoneme”, was inspired in part by the symbiosis of the two most common ANGIE configurations, letter-to-phoneme (also known as letter-to-sound) and phone-to-phoneme. The former has letters as the terminal units, while the latter casts phones as terminals. In both configurations, ANGIE models phonemics, aligning the phonemes in the pre-terminal layer with either phonetic sequences or word spelling transcriptions. In the letter-to-sound configuration, input word spelling information is lost as it is transduced into a sequence of phonemes. One factor driving the development of spellnames was precisely the desire to model phonemics while also capturing spelling information. The application of this approach to sound-to-letter modelling is obvious - if it were

---

<sup>1</sup>Henceforth we omit the quotes when using this relatively new term, for reasons of convenience.

possible to augment the sublexical hierarchy with a unit that fundamentally behaves like a phoneme, while at the same time containing spelling information, then perhaps the usual phone-to-phoneme configuration could be augmented to model spelling.

One difficulty in incorporating spellnemes into the ANGIE framework, for example by creating a set of units that directly substitutes phonemes in the modelling process, is the fact that there isn't a singular method to combine sound and spelling information into one unit. Work on low-level acoustic recognition and observations of the human sound production has produced a set of sound units of speech, called phones. Observations of word syllables and repeating sublexical patterns have yielded accepted sets of phonemes. No such explicit templates, inspired by physical observations, exist for spellnemes. This ambiguity implies that spellnemes need to be defined in terms of their relationships with existing sublexical units.

The first question is that of identifying the place of the new unit in the general sublexical hierarchy. In the context of the ANGIE multi-layer framework, one possible target for the combination of spelling and sound information is the terminal level - through augmentation of phones with spelling information. However, this approach is impractical because of the tremendous size of possible phone-letter pairs. Augmenting the terminal level with spelling information would shift the entire sound-to-letter task to the SUMMIT acoustic recognizer. However, the original SUMMIT task of transcribing acoustics into phonetic sequences is already quite difficult, as evidenced in part by the results obtained in this thesis (see section 5.3.7). Thus, it seems unwise to enrich SUMMIT's search space by substituting phones with "spellphones", which constitute a significantly more numerous set.

The principle of SUMMIT/ANGIE decomposition implied that SUMMIT was to be entrusted with the task of transcribing acoustics into phonetic sequences. Therefore, the effective input into the (remaining) sound-to-letter system would consist of SUMMIT phonetic transcriptions. Consequently, in the alignments on which the ANGIE column-bigrams are to be trained, SUMMIT phones form the terminal level. The upper sublexical levels in the ANGIE framework are fundamentally word-independent - it is impractical to augment the upper ANGIE syllabification

layers with spelling information specific to particular words, because the unit sizes become too big, and sparse data problems become severe. Therefore, any spelling information has to be encoded on the pre-terminal level, which is the phonemic level in the normal ANGIE framework. The above discussion is represented graphically in Figure 3-1.

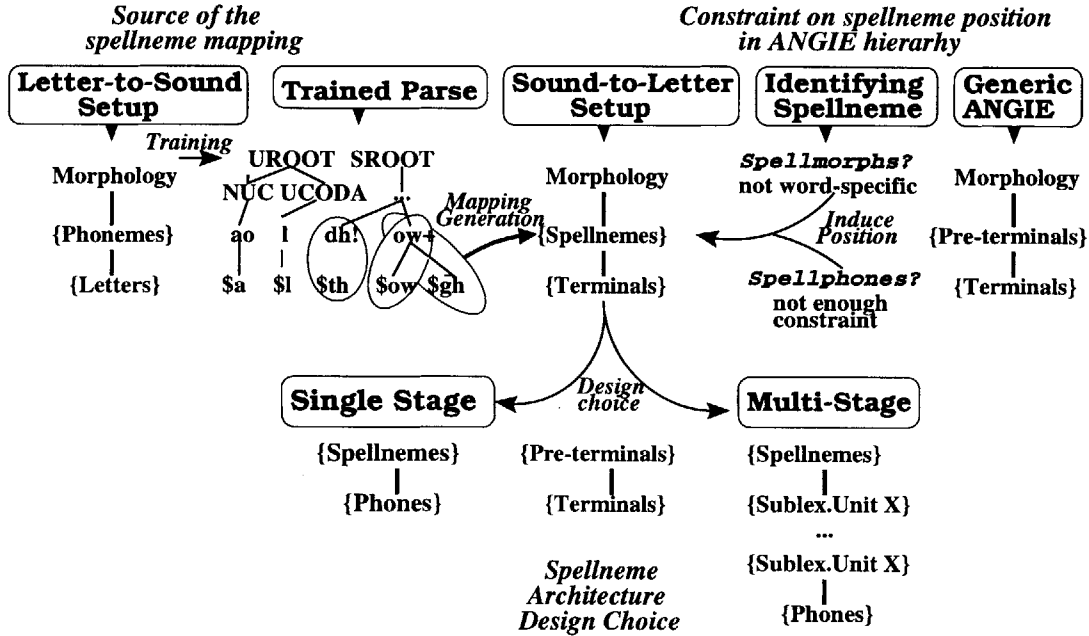


Figure 3-1: Deducing the spellneme-based framework: the top right part shows the constraints which defined the place of the spellneme in the sublexical hierarchy; the top left part describes how the set of spellnemes was derived from a common ANGIE configuration.

### 3.3 Design Tasks for the Spellneme-based ANGIE Configuration

The previous section motivated the notion that the place of the spellneme unit in the ANGIE hierarchy should be the pre-terminal layer, the usual position of phonemic information. Therefore, the sound-to-letter problem can be reformulated as the

problem of deriving spellnemes from phonemes by augmenting the later with spelling information. Two major tasks were identified as critical pieces in successful construction of an ANGIE system with spellnemes as pre-terminals:

i) Creation of the morph-to-spellneme lexicon.

ii) Creation of an exhaustive set of rules governing the role of spellneme units in the ANGIE framework.

Effectively, both of these tasks require the replacement of the phonemes with their spellneme counterparts - thus uncovering the underlying problem of defining the spellneme units and their mappings to phonemes, in the context of word spellings. The mapping that we were seeking can be described by the following functional expression: (letter, phoneme)-to-spellneme.

### **3.4 Letter-to-Sound: A Source for Spellneme Mappings**

The source for the (letter, phoneme)-spellneme mapping that we were seeking came from considering the logical complement of the sound-to-letter task. Indeed, the existing implementation of the ANGIE letter-to-sound recognizer implied that a set of sound-spelling mappings must become at least implicitly available after the system is trained. The ANGIE letter-to-sound system replaces phones with letters in its terminal layer, meaning that the alignment files contain simply the words and their spellings. The remaining upper layers in the sublexical hierarchy remain unchanged. During training, ANGIE aligns the most likely phonemes to a given sequence of letters. This alignment is precisely the source of possible (letter, phoneme)-spellneme mappings that is required for the sound-to-letter system. Indeed, given a phoneme, the set of all phoneme-letter alignments constitutes the set of all possible spellnemes – all possible “spellings” of particular phoneme. The logical outcome of this observation is to use the ANGIE letter-to-sound parses to fill up the (letter, phoneme)-spellneme mapping.

### 3.5 Parallel Construction of the Letter-to-Phoneme-to-Spellname map (LPSM)

During the course of our recognizer's development, we encountered the need to develop an efficient data structure that would model the mappings between phoneme and spellname units. This mapping was to be used in at least three important applications:

1. The expansion of a morph's phoneme sequence into the list of possible spellname counterparts.
2. The construction of the spellname-to-phoneme low-level grammar, for which it was necessary to determine all the phonemes mapping to a particular spellname (under all possible letter contexts).
3. The construction of the high-level grammars with spellnames taking the place of phonemes in the sublexical hierarchy: this task was the inverse of the problem in case 2, as it required to look up, for a particular phoneme, all the possible spellnames that it could yield in various spelling contexts, so that its entry in the high-level rules could be appropriately replaced.

The construction of the mapping was accomplished through an iterative process that built it up in parallel with the morph-to-spellname lexicon. The constructed spellname units were loosely based on a spellname subset proposed by Grace Chung [3], [2]. For her experiments in the Jupiter weather domain, Grace had constructed a certain morph-to-spellname lexicon - which became the starting point for the development of a set of spellnames that would cover the sublexical phenomena observed in the PhoneBook data. The principal entity in our construction was the (letter, phoneme)-spellname map. The original intention was to map a specific (letter,phoneme) pair to a unique spellname. However, it was discovered empirically that greater flexibility was required of this mapping. Some phonemes aligned with sequences of letters, such that, for a fixed first letter, several continuations for the remainder of the letter sequence were possible, depending on the parsed word. One example of this phenomenon is shown in Figure 3-2, which shows the ANGIE parse trees for the words

“corned” and “borrows”.

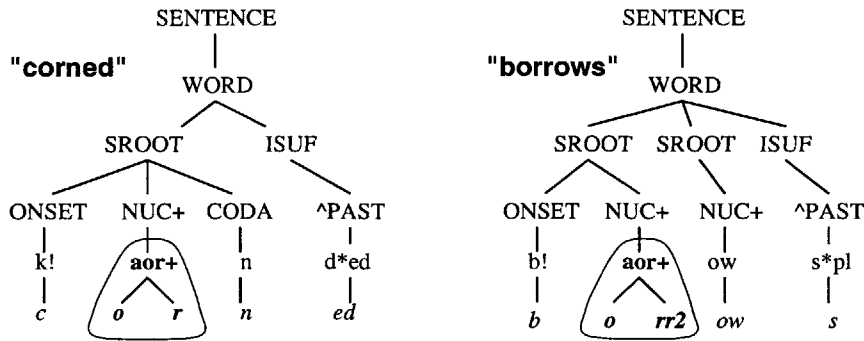


Figure 3-2: Example of an ambiguous one-to-many alignment for the phoneme *aor+*, in the context of the letter “o”.

For both words, the phoneme “aor+” is aligned with “o” as the first letter. However, it is not possible to associate a unique spellname to the (letter,phoneme) pair (“aor+,”o”) because, as the two trees show, at least two different spellings (“or”, “orr”) are possible for “aor+” in the context of “o”, depending on the continuation of the letter sequence (“r”, “rr2”).

Since such cases constituted a small minority of the observations, but were nevertheless significant, it was decided against expanding the map contextually to include letters surrounding the given alignment, and instead to allow the range of each mapping to overgeneralize as an OR-set of possibilities. Thus, a typical entry in the letter-phoneme-spellname map grew to include multiple range possibilities. Table 3.1 depicts the subset of LPSM mapping for the phoneme “aor+”, showing both normal and ambiguous mappings.

There were two main objectives in the construction of this spellname mapping: coverage and compactness. First, there was the question of determining all the possible spellnames that mapped to a given phoneme in the context of a certain letter. Second, it was necessary to prune all unnecessary mappings in order to avoid overgeneralization that could degrade final performance. The key step towards the realization of both of these goals was to link the problem of constructing the LPS mapping to the task of creating a morph-to-spellname lexicon. We proposed an iterative algorithm that gradually transforms the morph-to-phoneme lexicon into a morph-to-spellname

Letter	Phoneme	Spellneme
<i>Normal mappings</i>		
au	aor+	aur+
e	aor+	eor+
eo	aor+	eor+
o2	aor+	or+
oa	aor+	oar+
ou	aor+	our+
rr2	aor+	rr+
uo	aor+	uor+
<i>Ambiguous mappings</i>		
a	aor+	ar+,arr+
o	aor+	or+,ore+,orr+,ort+
oo2	aor+	oor+,oore+

Table 3.1: LPS mappings involving the phoneme **aor+**.

lexicon. On each iteration, the algorithm starts out by using the LPS map to replace morph-to-phoneme alignments with (possibly ambiguous) morph-to-spellneme expansions. These expansions are then stripped of phonetic labels to produce spellings, which are compared to the morphs. The spellneme sequences that yield correct spellings become part of the morph-to-spellneme lexicon. The remaining alignments are written out as errors. In the next step, the LPS map is incremented with the least possible number of mappings necessary to sufficiently expand the space of available spellnemes, and cover the sublexical space of (some of) the incorrect instances. After each iteration, the number of successful morph-to-spellneme lexical entries increases, and the number of failed parses decreases. In our experience, this iterative procedure produced a geometric decline in lexical errors. Thus, the parallel approach to spellneme lexicon and mapping creation transformed the daunting tasks of manually creating the 500+ element LPS map and a 11,000+ entry morph-to-phoneme lexicon into a tractable rapidly-converging problem.

In greater detail, the proposed iterative algorithm consisted of the following steps:

- Given: Morph-to-phoneme lexicon  $MP$ .
- Construction object 1: Morph-to-Spellneme lexicon  $MS = MP$ .

- Construction object 2: (Letter,Phoneme)-to-Spellname map  $LPSM$ .
- Set  $LPSM = \emptyset$ ,  $MS = \emptyset$ .
- Run modified ANGIE on the trained letter-to-sound domain:  
letters as terminals and phonemes as pre-terminals.
- For each proposed word-to-letter alignment:
  - For each morph:
    - Consider the sequence of aligned phonemes.
    - For each phoneme  $P$ :
      - Determine the corresponding terminal letter  $L$ .
      - Look up  $SP = LPSM(L, P)$ .
      - Write the set of spellname alternatives  $SP$  to the morph-spellname alternatives file.
- For each morph-to-spellname entry:
  - Create a tree of alignment alternative.s
  - Search the tree for a spellname sequence that matches the morph:
    - Build up a spellname sequence that at every step matches a substring of the morph.
      - If the entire morph has been matched,  
Return the matching spellname sequence.
      - Else backtrack and try a different spellname sequence.
    - If a matching sequence has been found,  
Write out the entry to  $MS$  lexicon.
    - Else write the morph and the alternatives to an  $MS$  errors file.
- For each entry in the  $MS$  errors file:
  - Consult the  $LPSM$  map.
  - If the error is due to a (letter,phoneme) pair with unaccounted spellname mappings, augment the map entry for that phoneme and letter with the necessary mappings.
- Repeat the ANGIE run, producing a new  $MS$  lexicon and errors file, and augmenting the  $LPSM$ , until all spelling errors are eliminated.

At each iteration, a new entry in the LPSM is created only when a need for a mapping arises from observing the actual ANGIE alignments. Specifically, modifications to the LPSM are made only when the algorithm fails to explain an alignment using the current mapping. In this way, only the relevant mappings are added to the LPSM, preventing overgeneralization and performance problems.

### 3.6 Multi-Stage ANGIE Configuration for Sound-to-Letter Recognition

One possible implementation of sound-to-letter recognition starting based on SUMMIT phones in the terminal level would involve the direct replacement of phonemes with spellnemes in the sublexical hierarchy (see Figure 3-1). While at first glance this approach seems like the natural application of the spellneme units, it was judged problematic because of the large increase in the low-level rules search space that would occur if spellnemes were used in ANGIE low-level rules. Indeed, a very important constraint that must be satisfied in training the ANGIE column bigram probabilities is the requirement that the number training utterances must be sufficiently large relative to the space of accepted parses that is given by the grammars. In the normal ANGIE framework, the low-level phoneme-to-phone grammar is already quite comprehensive, covering a very large space of phonological variants. Rewriting this grammar as a spellneme-to-phone mapping would cause the search space to balloon, as it would involve taking the product of the set of phonetic sequences and the set of spelling contexts: a particular spellneme would get mapped to all the phone sequences that were originally aligned with any of its phonemes – phonemes that mapped to that particular spellneme in some letter context.

It was thus decided to take a multi-stage approach to the spellneme-based sound-to-letter task. Instead of training up an over-generalized phone-to-spellneme recognizer on the usual word-to-phone alignments, we decided to keep the original ANGIE configuration of terminal phones and pre-terminal phonemes intact, and augment it

with a second ANGIE stage that would then transduce phonemics into our spellneme units. The advantages of this approach were numerous. First, we were able to avoid the problem of ballooning grammars, essentially replacing a mapping of cardinality [*Phones \* Phonemes \* Spellnemes*] with the more feasible [*Phones \* Phonemes + Phonemes \* Spellnemes*]. With this reduction, we were hopeful to avoid particularly critical overgeneralization problems. Second, retaining a well-developed ANGIE (phone-to-phoneme) configuration as part of the overall recognizer ensured that we had a solid basis on which to develop our multi-stage system, making it more modular, conceptually clearer, and easier to test and debug. Finally, the second ANGIE stage would serve as a higher-level language model, allowing us to test its influence on recognition separately from the lower-level parts of the recognizer. Hence the multi-stage approach simplifies the task of recognizer optimization, as it decouples various system modules and allows precise fine-tuning of the recognizer.

### 3.7 Low-level Rules Construction

The construction of the spellneme-to-phoneme grammar seemed, at first glance, a completely straightforward task, as it seemed to involve nothing more than taking a projection of the letter-phoneme-spellneme map to obtain all possible phoneme expansions for particular spellnemes. However, at this stage it was discovered that the existing spellneme-to-phoneme mapping was incomplete: certain phoneme units doubled as their own spellneme alternatives, into which they mapped by default (e.g. the phoneme “b!” mapped to the spellneme “b!”, in all contexts). The fact that these phonemes had exactly the same spelling as their spellneme variants, the iterative algorithm parsed words in which these were present correctly, and no entries was added to the LPS map as a consequence. Thus, when a projection of the map was taken to yield the spellneme-to-phoneme grammar, some of the most basic phonemes turned up missing. This caused the phoneme-to-spellneme FST to compose incorrectly with the phone-to-phoneme FST, which contained all the existing phonemes. Therefore, the phoneme-spellneme mapping had to be augmented - for every phoneme that did

not have spellname alternatives, a twin spellname was created, which mapped into itself when written into the spellname-phoneme grammar.

## 3.8 High-level Rules Construction

The construction of the high-level rules had to be performed manually, replacing the allowed phoneme expansions of the higher levels of ANGIE hierarchy with the possible spellnames.

## 3.9 Summary

In this chapter, we have introduced the “spellname”, a new sublexical unit that played a vital role in the proposed sound-to-letter framework by allowing introduction of spelling information and tightening of lexical constraint without severely affecting generalization capabilities. We motivated its lexical function by the need to model both phonetic and spelling information under the constraints imposed by the ANGIE sublexical unit hierarchy. Our first step was to identify the role of the spellname as the pre-terminal unit in the ANGIE hierarchy and define its functional relationship with the phoneme and letter units. Next, we presented an innovative algorithm that built up the LPS mapping and spellname lexicon in parallel, based on an existing letter-to-sound ANGIE configuration. Finally, we completed the design of the spellname-based ANGIE two-stage configuration, providing the motivation for the expected success of the proposed approach. In the next chapter, we turn to implementation details, seeking to illuminate the construction and operation of the overall sound-to-letter recognition framework.

# Chapter 4

## Implementation of the Recognizer

This chapter presents the architecture of our recognition framework, describing the various recognizer configurations, and identifying the engineering issues that were addressed during design. The development and operation of the proposed sound-to-letter recognizer can be decomposed into two main phases of construction and analysis. We proceed to describe each of these phases in detail, first by giving an overview of the component tasks and their inter-relationships, and then by presenting a complete account of each module.

### 4.1 Implementation Overview

The general flow of operation of the proposed SLR framework is illustrated in Figure 4-1.

The available PhoneBook corpus of utterances is split into several disjoint data sets. The training set is normally used to train the two ANGIE recognition stages - phone-to-phoneme and phoneme-to-spellname. Testing or development data are used for testing the recognizer's performance, which is analyzed in one of two possible ways. In the usual recognition configuration, the ANGIE stages are composed with the SUMMIT diphone-to-phone recognizer, producing an overall diphone-to-spellname configuration. The resulting recognizer is tested on input diphone sequences representing the (OOV) words in the test/development set.

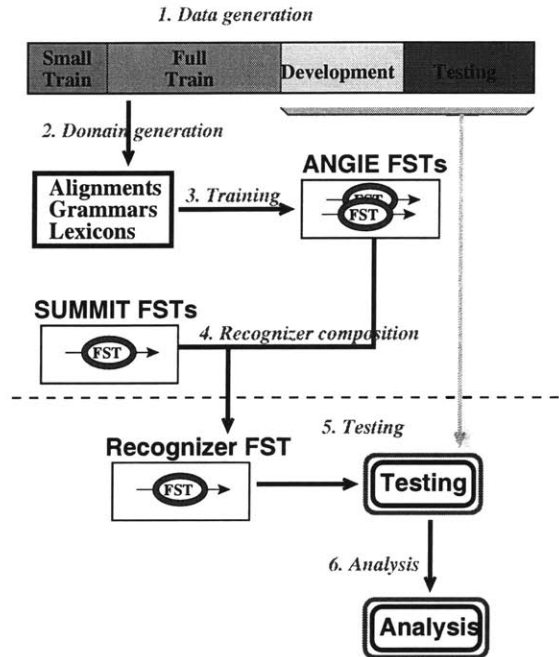


Figure 4-1: The overall sound-to-letter recognition framework

Secondly, in another batch of experiments, we considered the influence of (errors in) the SUMMIT acoustic recognition stage on the overall performance. In order to measure the effect of imperfect acoustic-to-phone recognition, we replaced the phonetic output of the SUMMIT recognizer stage with a “forced” phonetic sequence for each test word, obtained by running the recognizer in such a way that it is constrained to recognize only the single word that was spoken. “Forced” in this context thus implies that the obtained phonetic sequence is the best possible phone sequence for the given word. By comparing forced mode recognition results with those obtained using raw acoustic data, we were able to evaluate the phone-to-spelling performance separately from the acoustics-to-phonetics aspect.

## 4.2 Recognizer Construction

The overall progression of recognizer construction is shown in Figure 4-2, and consists of the following four basic steps.

1. Data generation: Phonebook domain, training/test/development sets.

2. Domain generation: generation of ANGIE alignments, grammars.
3. Training: training ANGIE, construction of column-bigram FSTs.
4. Recognizer construction: weighing and composing the resulting FSTs.

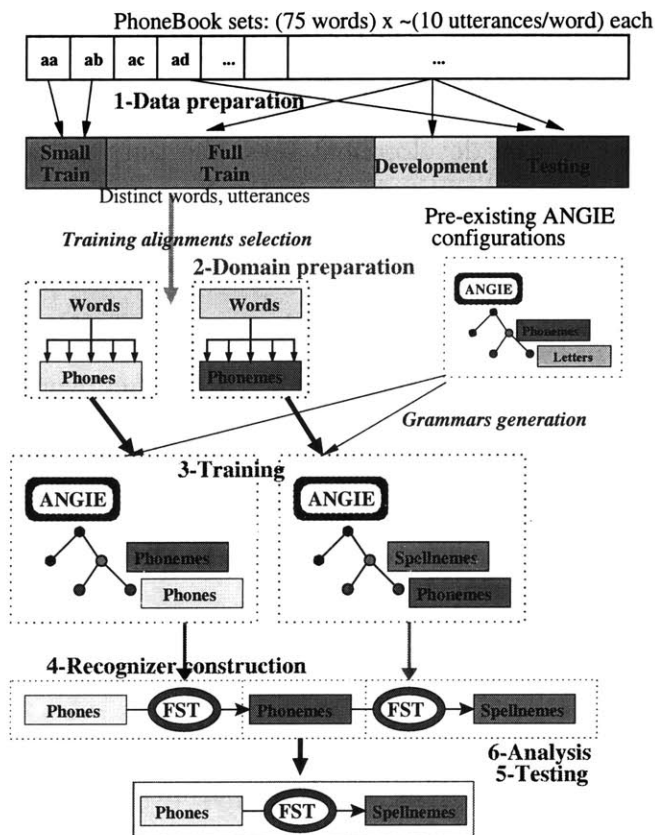


Figure 4-2: Stages of recognizer construction.

In the data preparation step, the PhoneBook speech data were separated into disjoint training, development, and test sets, as mentioned above. Each set contained an independent set of words gathered from an independent set of speakers. Domain preparation was the most effort-intensive step in our development, as it involved extensive modification of existing ANGIE systems, manual preparation or editing of certain grammars, and other tasks described in Chapter 3. Training of the ANGIE recognizer and the subsequent construction of column-bigram FSTs relied heavily on existing SLS tools, partially introduced in Chapter 2. Similarly, the composition of FSTs presented no new technological challenges; however, the optimal weighing of

the component FSTs was a major optimization challenge that greatly influenced the recognition architecture.

### **4.2.1 Data Preparation: PhoneBook Corpus**

We begin by describing PhoneBook – the corpus of utterances used in the development of the proposed sound-to-letter recognizer. Then, we explain the division of the speech data into training, testing and development sets, the purpose of the sets and their general usage in our experiments.

The PhoneBook corpus is a collection of isolated-word telephone speech utterances, designed to represent a phonetically-rich set of words [5]. It consists of a total of 93,667 isolated-word utterances, spanning 7979 distinct words, each said by an average of 11.7 talkers, with 1358 talkers each saying up to 75 words. All data were collected in digital form, via a T1 telephone line, from a demographically representative sample of adult native speakers of American English. There are 106 word lists used in the corpus, and these are identified by two-character sequential alphabetic codes: "aa", "ab", "ac", ... "eb". In our experiments, these word sets have been separated into separate training, development, and testing sets; further, a subset of the full training set is used for running smaller experiments designed to yield preliminary findings that can later be verified on the full training set (also referred to as "fulltrain" in our experiments); henceforth this smaller set will be referred to as the "small training", or "smalltrain" set.

### **4.2.2 Training, Development and Test Sets**

The purpose of the data preparation step is to formalize the division of available speech data into training, development and test steps. This split directly affects the performance of the recognizer and the validity of recognition results. The bulk of the speech data is reserved for the training set, which amounts to 85 percent of available utterances. The remaining 15 percent is split roughly evenly between the development and the testing sets. The main purpose of the development set is

recognizer optimization. In our recognition framework, many different configurations are considered, and the performance of each one is tested on the development set. Comparison of relative performances on the development set allows us to identify the best possible recognizer configuration. However, the development set may not be the correct indicator of the performance of the optimal recognizer – there is a possibility that, because of data or training idiosyncrasies, the chosen recognizer only performs well on the specific utterances in the development set. Thus, another set of utterances, the test set, is used to verify the performance stability of the optimal recognizer. The test set is considered to be the final, unbiased indicator of the recognizer’s performance.

The basic format of the experiments reflects the division of PhoneBook data into small and full training sets, a development set, and a test set. Normally, the ANGIE grammars were trained on the full training set, and then the resulting recognizer configuration was tested either on the development (for parameter optimization) or the testing (for final analysis) set. Many initial tests were carried out using recognizers trained on the small training set, either for the purpose of (i) debugging the components of the testing script, or (ii) probing the recognizer’s performance under a given set of conditions, e.g. obtaining preliminary performance numbers for a particular FST weight configuration. Finally, in order to gauge the generalization capability of the recognizer, several experiments were run using the PhoneBook training utterances in the role of a recognizer test set.

### 4.2.3 ANGIE Domain Preparation

Chapter 2 introduced the concept of an ANGIE domain, and Chapter 3 described the essential procedures used to construct the phoneme-to-spellname mappings. The goal of this section is to describe the tasks of preparing the ANGIE domains for the various stages of our recognizer.

The first important facility required for the creation of various recognizer configurations, i.e. ANGIE domains, was the alignment-generation mechanism. The two ANGIE stages yielded two separate alignment sets, word-to-phone-sequence and

word-to-phoneme-sequence. The division of the data into training and test sets for various experiments meant that the alignment data had to be decomposed into disjoint parts.

In our implementation, the need to generate various alignment sets induced the creation of database-like data structures to keep track of words, their alignments, and locations of their utterances in the Phonebook corpus. For brevity, we omit the details.

The generation of ANGIE grammars and lexicons for the phoneme-to-spellname stage was described in Chapter 3 – recall the letter-phoneme-spellname map (LPSM) and the iterative algorithm for updating the map in parallel with the morph-to-spellname lexicon. On the other hand, the grammars and lexicons for the phone-to-phoneme stage were already available from existing SLS ANGIE domains, and required little, if any, modification.

#### 4.2.4 ANGIE Training and FST Construction

The training of the two ANGIE stages is a two-step process. In the first training step, the ANGIE grammars are trained up based on input alignment files and low-/high-level grammar rule sets. In the second step, the observed parse trees are converted into column-bigram FSTs. The training of ANGIE grammars and creation of column-bigram FSTs was combined into a single (first) phase, handled by the script `train-slr.cmd`. The resulting FSTs are saved to a repository for future use in recognizer compositions, as described in more detail below. Since the sound-to-letter recognizer has two ANGIE stages, it was logical to perform the training for both stages in a single step handled by the script, as illustrated in Figure 4-3.

Here is an example of such a command:

```
train-slr.cmd -phbk-w2p-set smalltrain 60000 -phbk-w2ph-set alltrain
7500 -angie-or-cb both
```

This command specifies the following:

**Stage 1: ANGIE/Column-bigram training: `train_slr.cmd`**

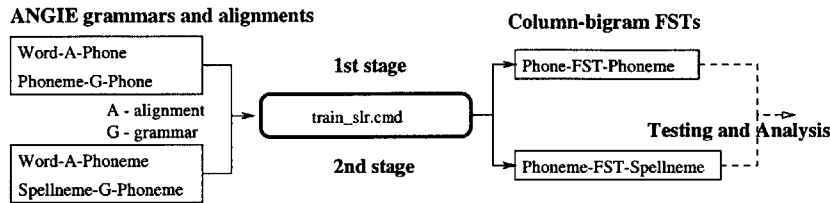


Figure 4-3: Recognizer training: `train-slr.cmd` schematics

- Train the 1st stage phone-to-phoneme ANGIE grammar and construct the FST based on the small set of training alignments, using the first 60000 alignments.
- Train the 2nd stage phoneme-to-spellname ANGIE grammar and construct the FST based on the full set of alignments, using the first 7500 alignments.

#### 4.2.5 Reusability of Recognizer Components

Reusability of recognizer components was a major goal pursued in the design of the recognizer framework, in the interests of both simplicity and speed. In particular, the sequential nature of recognizer creation implies that the intermediate building blocks can be saved and stored after they are first created, in order to be reused at later stages. To simplify matters, in our system, there were just two relevant dimensions to FST training – the sizes of the training sets for the two ANGIE stages. The major phases of recognizer training that are relevant to the concept of reusability are as follows:

1. Alignments: Creation of alignment files.
2. Grammars: Training the ANGIE grammars given the alignment files.
3. FST: Construction of the ANGIE column-bigram FSTs from the grammars.

In our implementation, each of the first three steps produces components that can be reused in the successive stages, ultimately resulting in the construction of the full recognizer. These intermediate components – alignment files, grammars, and FSTs, are stored as files. In order to access existing components, we've designed a naming system that builds up the name of a file based on the characteristics of its

content. The file name is automatically generated by the system. The reusability paradigm was put into practice in two successive steps. Alignment files for both the small and full training sets were generated and reused for future column-bigram creation; the column-bigram FSTs for various training domains were likewise generated once, and reused in future compositions. For example, a word-to-phoneme alignment file for the second stage ANGIE FST based on words from the full training set, would have the name `'word2phoneme.align.fulltrain'`. In the next stage, if the `train-slr.cmd` script is run with the “fulltrain” option, the correct alignment file will be accessed for the training of ANGIE grammars. The same procedure is used in the remaining training steps, with script options identifying the appropriate files. Component reusability helped us reduce both the complexity and the execution time of the training stages.

### 4.3 Recognizer Evaluation

In evaluating the recognition performance of the proposed sound-to-letter framework, we reused the FSTs obtained in the recognizer training phase. Our performance analysis framework supports multiple recognition runs over various recognizer configurations. For each particular configuration, this phase can be further decomposed into three stages:

1. Recognizer creation: weighing and composing the resulting FSTs.
2. Testing: running utterances through the recognizer.
3. Analysis: compiling recognizer performance information.

The testing step was the most time-consuming step of the experiments, since for some configurations it took several hours to complete on the fastest available machines. Finally, the extraction of performance information and its analysis necessitated the creation of several new and innovative engineering solutions.

### 4.3.1 Batch-run Architecture Overview

One of the most appealing properties of the multi-stage sound-to-letter solution is the ability to change the relative influence of each of the ANGIE recognition stages on the overall performance. Specifically, the weight assigned to arc-transition probabilities of each of the two ANGIE FSTs can be changed. For example, assigning a weight of zero to the language model phoneme-to-spellname FST implies that the phoneme-to-spellname grammar will impose no probabilistic constraint on recognition hypothesis, and will only restrict the most likely phonemic sequences given by the phone-to-phoneme stage to the space of possible spellname sequences (clearly, such a configuration would not be expected to attain good performance). One of the immediate goals of our proposed experiments was to deduce the optimal weight configuration for the multi-stage recognizer. With this goal in mind, we proceeded to design and implement a recognizer architecture that readily supported construction and testing of various recognizer configurations. The variable-configuration testing framework is illustrated in Figure 4-4.

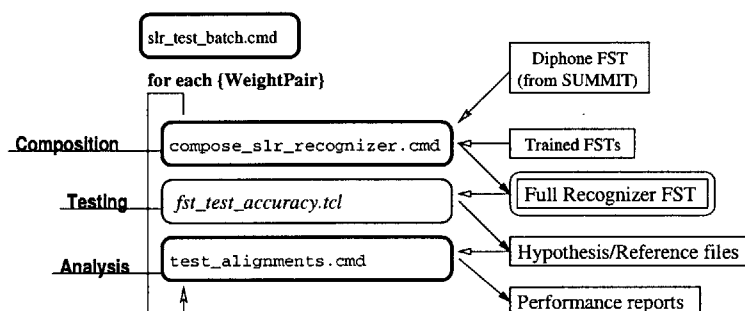


Figure 4-4: A brief flow-chart of recognition phases.

The batch-run configuration is discussed in more detail below.

### 4.3.2 Recognizer Construction for Normal and Forced Recognition Experiments

Figure 4-5 summarizes the two main recognizer configurations - normal recognition (unknown words) and forced phonetic recognition (forced “correct” input phonetics).

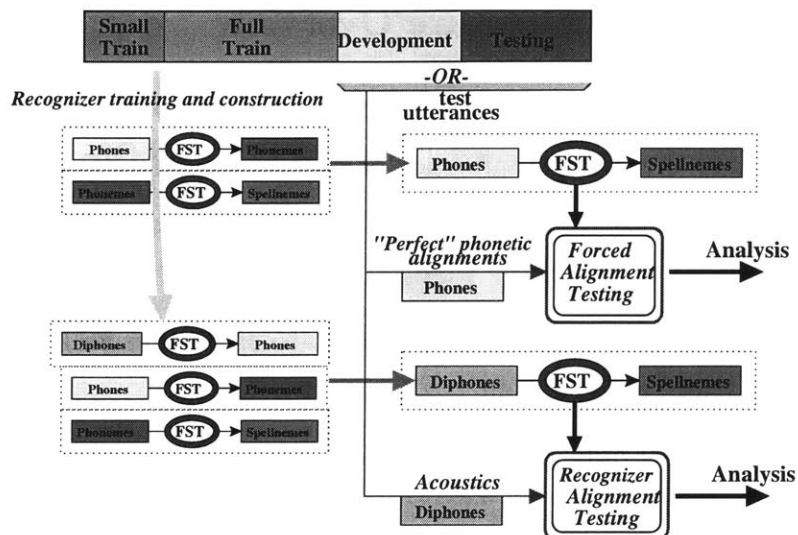


Figure 4-5: The two main recognizer testing configurations: Forced – phoneme-to-spellname recognizer (Top) and Normal/Acoustic – phone-to-spellname recognizer (Bottom).

In normal recognition mode, the two ANGIE phone-to-phoneme and phoneme-to-spellname stages are joined to a SUMMIT diphone-phone FST to form a complete recognizer that parses acoustics of unknown utterances. In the forced recognition mode, the composition with the SUMMIT FST is replaced by a set of “correct” phonetic sequences that are fed directly into the first ANGIE stage.

Ensuring a seamless composition of the phone-to-phoneme and phoneme-to-spellname FSTs involved several issues. One technicality lay in ensuring compatibility between the phones produced by SUMMIT and the terminal phones used in the phone-to-phoneme ANGIE FST, as some discrepancies were found between the phones written out in alignment files, and subsequently between the resulting phone-to-phoneme FST (dash-r) and the phones in the SUMMIT FST (-r).

The second issue involved reconciling the parsing of functional classes (upper level sublexical units: **sroot**, **uroot**, **dsuf**, **isuf**, **pre**, **spre**, **fcn**. During normal transformation of ANGIE column bigrams into an FST, these functional classes are retained from the best-fit hypothesis, and become part of the recognition output. However, since in our approach, the phone-to-phoneme and phoneme-to-spellname FSTs were composed together, the output of functional classes from the phone-to-phoneme

ANGIE FST necessitated modifying the phoneme-to-spellname FST to transduce these functional symbols (into themselves).

### 4.3.3 Testing the Recognition Framework

The overall schematic of our main evaluation tool, the *slr-batch-test* command script, is shown in Figure 4-6.

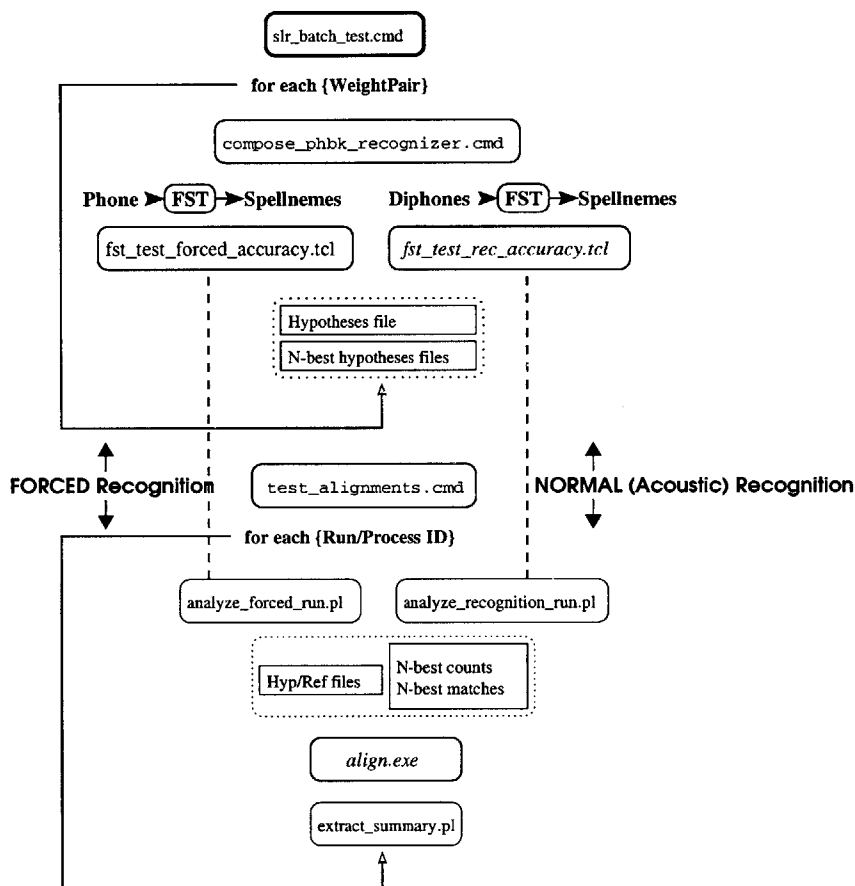


Figure 4-6: Recognizer Evaluation: `slr-batch-test.cmd` schematics.

Testing the normal recognizer configuration was a fairly straightforward operation based on the *fst-test-accuracy* SLS script. This tool can be configured to take a number of arguments:

1. An FST with diphone inputs (such as our sound-to-letter recognizer).
2. A control file that specifies the PhoneBook sets to be tested on.

3. Number of N-best hypotheses to keep.

Given these arguments, the *fst-test-accuracy* tool runs input utterances from the requested PhoneBook set through the recognizer, and outputs the transduced units (including N-best lists, if requested), which in our case are the spellnemes. In the forced phonetic alignments task, the evaluation procedure for our recognizer had to be modified. Recall that in the forced mode, the SUMMIT diphone-to-phone FST stage was replaced by “correct” phonetic sequences. The first task was to create the word-to-phone alignments for the development and testing sets. Then, given the set of alignments and the phone-to-spellneme FST obtained from the two ANGIE stages, the following sequence of steps was performed:

For each word-phones alignment in the testing file:

1. Create an identity FST from the string of phones.
2. Compose the FST from Step 1 with the phone-to-spellneme.
3. Project the N-best output of the composition - a string of spellnemes.
4. Evaluate the FST performance as in the normal recognition case.

#### 4.3.4 Analyzing the Recognition Performance

The results obtained from analyzing the performance of the SLR recognizer are described in detail in Chapter 5. We begin by describing the methods that we used to assess recognition performance.

The main tool used to analyze recognizer performance was the *align* testing tool. Given two files - one with hypothesized sentence spellings, and one with the reference sentences, this tool generates a detailed summary of sentence and word accuracy, complete with insertion, deletion, and substitution statistics both for the entire set and individual words. This tool also generates statistics for letter confusion pairs, which help isolate possible sources of errors in the recognizer. In our experiments with isolated words, the words played the role of sentences, and the letters were considered as words.

### 4.3.5 Batch-run Implementation

The design of the recognizer's testing and analysis stage, associated with the `slr-test-batch.cmd` script, was driven by two main factors: (i) the need to optimize recognizer performance over a large set of FST weight pairs, and (ii) the existence of several idiosyncrasies in the available FST-testing tool. Each run of `slr-test-batch.cmd` can be thought of as a test of our recognition system with respect to particular training and testing sets. Moreover, within each run, several possible recognizer configurations are possible, corresponding to different weighings of component FSTs. We configured this shell script to support running several experiments in order to optimize parameter settings and measure performance. An example of a specific run will help to elucidate the discussion of the procedure. Consider the following instantiation of the command file:

```
slr-test-batch.cmd -forcedPaths -phbk-test-set development -phbk-train-set
smalltrain alltrain -keepHypRef -keepCounts -keepSums -write-perf
-nbest 50 -weights 0.3 1.0 0.5 1.0 0.7 1.0
```

The following options are thus specified:

- The recognizer will be configured for the forced recognition mode, meaning that only the two ANGIE FSTs will be composed (with the specified weights), resulting in an FST that will accept phone sequences at the inputs.
- The FST used for the first stage will be the one trained on the `smalltrain` alignment set; for the second stage, the complete `alltrain` FST will be used. Based on the designations *smalltrain*, *alltrain*, the FSTs produced by the preceding training run will be used.
- The performance of the recognizer will be tested on the development set: word-to-phone alignments will be generated for all words in the development set, pushed through the recognizer, and the results evaluated.
- Three separate recognizers will be composed and tested based on the column-bigram FSTs given by the first option. The three weight pairs used to construct

the recognizers are (0.3,1.0), (0.5,1.0), and (0.7,1.0).

- Testing of the specified recognizer configurations will include information on the 50 N-best hypotheses for each parsed utterance.
- The control file that specifies testing parameters, including the set of PhoneBook utterances for recognition runs, is `phbk-test.espec`.
- For analysis purposes, recognizer hypotheses for each word in the testing set will be kept, and the depth of the correct (if any) hypothesis in the N-best list will be recorded.

## 4.4 Implementation Languages

The technical centerpiece of this thesis was a body of functions and data structures used to represent and manipulate the various inter-unit relations of the ANGIE framework - grammars, maps, alignments. In order to keep consistent with existing ANGIE architecture, it was decided at the outset to implement all additions to the system in C.

Another significant engineering challenge lay in integrating the numerous executables and scripts used in our project. These routines handled a wide variety of tasks, including training ANGIEs, constructing column-bigrams, modifying, composing and testing FSTs, and analyzing recognition hypotheses. The glue that was used to organize the symbiotic operation of various modules was the C-shell command file environment. Fortunately, our recognition experiments naturally decompose into a sequence of stages, such as training, testing, and analysis. In our implementation, each logical stage was encapsulated by a C-shell script that grouped together a sequence of related routines/function calls.

Several Perl scripts were created, mainly for the tasks of extracting needed information from files generated by other scripts and executables. This information was then re-packaged for use by successive modules. A file-to-Perl-to-file paradigm was used to bridge the I/O gaps between available and new routines that were written at different times with different usage procedures in mind.

Finally, Matlab was the natural language of choice for the analysis and representation of recognition results.

## 4.5 Summary

In this chapter we have presented the architecture of our recognition frameworks, as well as the main methods used in testing and analysis. Thus, the stage is set for a thorough evaluation of our multi-stage design, which is the subject of Chapter 5.



# Chapter 5

## Recognizer Evaluation

The goal of this chapter is to describe in detail the experiments that were performed in the course of this research, and to present the relevant results and findings in a unified framework.

The experiments that were carried out on the proposed sound-to-letter recognizer can be divided into two basic phases. The goal of the first, preliminary phase was the optimization of the recognizer's parameters. The optimal recognizer configuration thus obtained served as the test-bed for the main body of experiments carried out in the second phase. Thus, the purpose of the optimization stage was to identify the best possible recognizer configuration, both in order to obtain the best possible recognition performance, and to ensure that any performance issues were not caused by poorly chosen system parameters. The objective of the following phase was to evaluate the performance of the optimized recognizer configuration. The account of the second-phase experiments given below presents a logical sequence designed to give a complete account of recognition performance and the effect of various modules on recognition accuracy.

### 5.1 Performance Measures

In our evaluation of the proposed sound-to-letter system, we relied on several common performance indicators that describe the quality of letter-sequence prediction, based

REFERENCE	f	r	A	g	m	E	N	t	A	l	*
HYPOTHESIS	f	r	E	g	m	I	T	t	*	l	E
Errors			S				S	S	I	D	
Substitutions (SR)	=	30.0	Correct (LAR)				=	60.0			
Deletions (DR)	=	10.0	Errors (LER)				=	50.0			
Insertions (IR)	=	10.0									

Table 5.1: Recognition hypothesis/reference alignment for the word “fragmental”. (S)ubstitution, (I)nsertion, and (D)eletion errors are listed below their occurrences, followed by the error rate statistics for the word.

on the comparison of the reference words with the spellings extracted from the top recognition hypotheses. Specifically, the hypothesized spellings were aligned with the reference words in a way that minimized the number of letter substitutions (S), insertions (I) and deletions (D) necessary to transform the hypothesis into the correct spelling; then, the S/I/D counts for each of the evaluated words were combined into error rates for the overall test set. The following is a summary of the error rates (in the context of the operations needed to transform the hypothesis into the reference word). A typical reference/hypothesis alignment, together with the accompanying error statistics, is given in Table 5.1.

1. Substitutions rate (SR): rate of replacement of letters in the spelling hypothesis.
2. Insertions rate (IR): rate of insertion of additional letters into the spelling hypothesis.
3. Deletions rate (DR): rate of deletion of letters from the spelling hypothesis.
4. Letter error rate (LER): the overall error rate  $\Rightarrow LER = SR + IR + DR$ .
5. Letter accuracy rate (LAR): the percentage of correctly identified letters in the reference  $\Rightarrow LAR = 1 - (SR + DR)$ .

## 5.2 Recognizer Optimization

In the context of the proposed multi-stage recognizer, the primary optimization task was that of assigning proper relative weights to the two ANGIE FSTs resulting from the phone-to-phoneme and phoneme-to-spellname recognition stages. One attractive feature of FST modelling is the possibility of scaling the state transition probabilities of any one of the FSTs participating in the composition. This has the effect of either enhancing or decreasing the influence (weight) of a particular FST stage on the overall transducer. The main objective of our recognizer optimization phase was precisely the identification of the optimal weight combination for the component ANGIE FSTs.

The major difficulties associated with FST weight optimization task were the massive size of the possible search space, and the testing speed of an individual recognizer configuration. The naive way to approach the task of recognizer optimization would be to consider the largest possible set of weight combinations, and then for each weight combination, train up the recognizer on the full training set, and test it on the full development set. Unfortunately, several problems preclude the possibility of such a brute-force approach. First, there are no obvious limits on the (relative) weight of the FSTs, and there is no available *a priori* information about recognition performance as a function of these weights. Thus, one can not accurately preset either the range of FST weight combination to check, or the size of the weight increments at which the resulting recognition configurations are to be evaluated. The second problem is the time that it takes to test the development set performance of a particular configuration. For a recognizer trained on the full training set, the evaluation of the proposed sound-to-letter recognizer on the test set consumed almost four hours. In the presence of pressing time constraints, the length of a full testing run thus required a more thoughtful approach to recognizer optimization.

In the initial tests, the tested recognizer was trained on the full PhoneBook training set and evaluated on the first 1000 utterances of the development set. The resulting Letter Accuracy Rate (LAR) performance for various FST weight combinations is shown in Figure 5-1.

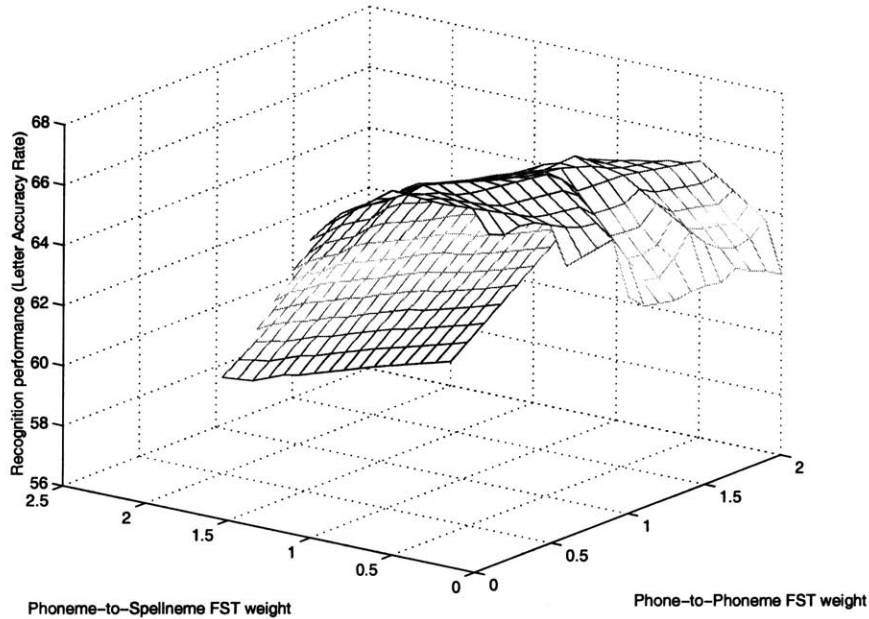


Figure 5-1: Recognition accuracy (LAR) as a function of component FST weights. The performance of the fully-trained recognizer, evaluated on the first 1000 utterances of the development set.

Several interesting observations can be made from these testing runs. First, the preliminary optimal weight configuration appears to be  $(0.3, 1.0)$ , meaning a weight of 0.3 for the phone-to-phoneme FST and 1.0 for the phoneme-to-spellname FST. Second, recognition performance appears to be a fairly smooth function of FST weight pairs. The continuity of the function shape supports the assumption that the relative recognition performance of different weight combinations will remain constant with respect to changes in other system variables, such as training and test set properties. This furthers our hope that the assumption about constant relative performance of recognizers built from two different FST weight pairs is true. Moreover, the apparent smoothness of the mapping attests to some stability in the results, as well as to the stability of the overall system. Henceforth, we consider the weight pair  $(0.3, 1.0)$  optimal, and use it for subsequent recognition experiments.

## 5.3 Recognition Experiments

The general format of the recognition experiments reflects the separation of the PhoneBook data into training, development and test sets. Normally, the recognizer whose performance is to be analyzed is trained on the full training set: this means that both ANGIE stages are trained on alignments of words in the PhoneBook training set; subsequently, they are tested on utterances from either the test or development set. However, there are some exceptions to this setup. For example, in order to test the recognizer’s capability to generalize the lexical patterns that it has learned, it is tested on the training set to provide a basis for comparison. In another scenario, the second ANGIE phoneme-to-spellname FST stage is trained on all available alignment data to simulate the case when the phonemics of all possible words are known; however, the first stage ANGIE FST is still trained only on the training set.

### 5.3.1 Overview of the Experiments

The following is a brief overview of the succession of experiments performed in this thesis. All the experiments utilize the optimal recognizer configuration, based on FST composition weight pair (0.3, 1.0). We begin by testing recognition performance, on the development set, of the optimal recognizer configuration, as discovered in the previous section. In the second experiment we explore the generalization ability of the recognizer, evaluating its performance on the training and test sets. Next, we test the generalization performance of the the second ANGIE phoneme-to-spellname stage, training it on phonemic data from all existing words. In the next two experiments we test the effect of individual recognizer stages on recognition performance. First, we determine the benefit of employing the ANGIE second stage “language model” - the subject of detailed development in Chapter 3, by testing recognizer configurations with all weight assigned to the first phone-to-phoneme stage. Second, we test the extent to which the recognizer performance is degraded by mistakes in acoustic recognition by replacing the outputs of the SUMMIT stage with perfect phonetic alignments. These “forced” phonetic recognition (FPR) experiments also

serve as an important sanity check for the validity of the ANGIE-based multi-stage approach, providing information about the coverage attained by the proposed recognizer. The training and testing sets used in these experiments mirror those used in regular recognition mode. Finally, in order to assess the consequences of unobserved column-bigram patterns, we explore the utterance instances for which the recognizer was unable to find the right hypothesis in forced phonetics mode, and examine the differences in overall recognition performance between the failed words and the ones that were covered by the observation space.

### 5.3.2 Optimal Configuration Performance

Recall from Section 5.2, that the optimal weight configuration for the proposed sound-to-letter recognizer is (0.3,1.0). The baseline performance for our recognizer was obtained by training the two ANGIE FSTs on the full training set and evaluating the performance of the resulting recognizer on the test set. We obtained a 41.0 percent overall error rate (LER) and a 69.5 percent letter accuracy rate (LAR).

### 5.3.3 Generalization Performance

One of the most significant indicators of a recognizer’s performance is its ability to generalize well - that is, exhibit high performance on unobserved sequences. Generalization performance is usually checked by comparing the recognition performance on an unobserved test set with the baseline performance on training data. A recognizer that generalizes well - one that has truly “learned” the patterns in the training data, is going to perform nearly as well on the unknown words as it did on its training examples. In our case, generalization performance can be measured in regard to three sets: training, development and testing. We expect the SLR to perform best when evaluated on the training set - words that it has “seen” in training. If, in training, the column-bigrams have adequately captured their underlying mappings, we would expect a relatively small degradation in recognition performance when the SLR is evaluated on the development set. Finally, given the possibility that the proposed

Evaluation set	SR	DR	IR	LER	LAR
Smalltrain	16.6	7.5	10.0	34.1	75.9
Development	23.1	9.6	11.7	44.4	67.3
Test	21.2	9.3	10.5	41.0	69.5

Table 5.2: Generalization performance of a fully-trained recognizer. The LER indicator is the Letter Error Rate, combining substitution, insertion, and deletion error statistics.

“optimal” configuration had accidentally performed well on the specific development set data, another smaller degradation error could be expected when evaluating test set performance. Table 5.2 shows the generalization results for the “fully-trained” recognizer.

Thus, there is a 20.2 percent increase in Letter Error Rate when the recognizer is evaluated on the development set, followed by a 7.7 percent LER decrease on the test set. Thus, there is a significant performance penalty for OOV letter recognition, propelled mostly by the 39.2 percent increase in substitution errors. However, the fact that the Letter Error Rate is smaller for the test set than for the development set is an indicator of the absence of recognizer bias toward the development set utterances. This assertion reaffirms the optimality of the chosen recognizer configuration.

### 5.3.4 Hypothesis N-best List Statistics

The capability to generate N-best hypothesis lists from the proposed sound-to-letter recognizer inspired the following experiment, whose purpose was to compare recognition coverage and accuracy between variously trained recognizer configurations. Specifically, for each evaluated utterance, we searched the corresponding recognition N-best hypothesis list for the correct hypothesis (correct spelling), and recorded the depth (if any), at which it was found. For example, if, for a given pronunciation of the word “achieve”, the first instance of a matching spelling hypothesis was the 10th-best alternative, the count of hypotheses found at depth 10 was incremented by 1. Accumulating the counts across all tested utterances, we were able to obtain

a picture of how well a given recognizer configuration finds the correct hypothesis somewhere in its search space.

Figure 5-2 presents the results of our findings. We recorded the N-best counts for four recognizer configurations, identified by the level of 1st and 2nd stage ANGIE training. The configurations were evaluated on the development set. For example, the point in the graph denoted by *X* makes an assertion about the performance of a recognizer whose phone-to-phoneme ANGIE phonological model is trained on the small training set, and the phoneme-to-spellneme stage is trained on all word-phoneme alignments. Thus, we see that 1000 correct answers, or about two-thirds of the total correct hypotheses found by this particular configuration, have occurred no lower than the 10th best alternative.

Several observations can be made about performance described by Figure 5-2. First, the overall coverage of correct hypotheses leaves something to be desired for even the best, most-trained configuration. From 7000 development set utterances, only in roughly 2500 instances, did the recognizer include the correct answer among its top 50 hypotheses. Second, there is no clear N-best cutoff beyond which the remaining hypotheses can be called useless; although tentatively, one can set 25 or 30 as the number of best hypotheses needed to capture most of the correct answers (if any). A reassuring property of the graph is the regularity of N-best depth performance with regard to the training of evaluated configurations. Both configurations trained on the full training set outperform the configurations trained on the smaller training subset. Further, an increase in second stage training, holding first stage training constant, results in a definite performance increase.

### 5.3.5 Generalization performance of the Phoneme-to-Spellneme ANGIE second-stage

We have identified two main unknown word contexts in which our system's performance could be evaluated. These contexts give different flavors of defining OOVs, and offer a trade-off between simplicity and performance. The first effectively defines

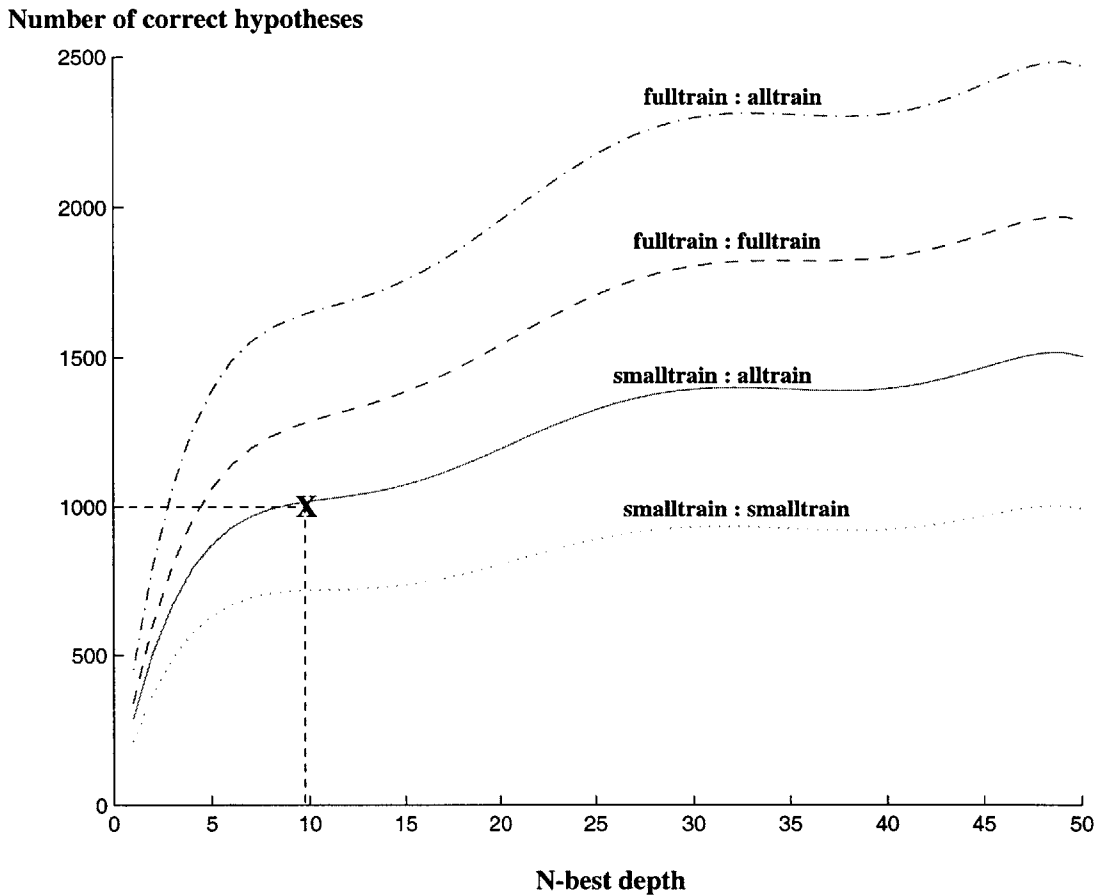


Figure 5-2: Cumulative N-best depth counts for correct hypotheses. Each curve corresponds to a particular training configuration (1st stage training set : 2nd stage training set). A particular point on a curve describes how many correct recognition hypotheses (Y-axis) appeared in the recognizer's N-best list before or at the given depth (X-axis). For a total of 7500 words, at most 2500 correct hypotheses were found, given the top-performing configuration and a maximum N-best depth of 50.

OOV words as ones which are not in any way known, while in the second context, OOV words are those whose pronunciation has never been encountered, but for which phonemic transcription is available.

According to the simplest definition, an out-of-vocabulary word is one that has never been seen before, in any form, by the recognizer. The separation of PhoneBook data into disjoint training and test/development tests corresponds to this definition. Normally, we assume that every testing word is OOV, and that absolutely no sublexical information is known *a priori* about the word, aside from what can be inferred by observing other words.

Evaluation set	SR	DR	IR	LER	LAR
Development	22.7	9.4	11.6	43.7	67.9
Test	20.8	9.1	10.4	40.3	70.1

Table 5.3: Generalization performance of a recognizer with a complete language model.

For real-life applications, however, we can imagine that the phonemics of most words can be known in advance - for example, given by an existing lexical dictionary. For example, the PhoneBook corpus includes phonemics for each word. At the same time, we make no claims about knowing about the actual word pronunciations (phonetics). Subsequently, we can investigate the performance of a recognizer whose phonetic stage is trained on the usual subset of alignments, as before, but whose phonemic language model is fully-known. In other words, we assume that phonemic alignments are available for the OOV words, for example by being looked up in an extensive dictionary. For our two-stage recognizer this translates into training the second phoneme-to-spellname stage on all of the PhoneBook word-to-phoneme alignments, i.e. augmenting the training lexicon with full lexical coverage of the development and test sets. The goal of the next experiment was to measure the generalization performance of the spellname-based ANGIE second stage. If the second stage grammar captures the training data patterns well, we would expect the addition of test alignments to the training set to have a small effect on the overall performance. In this experiment, we repeated the sequence of evaluations for the normally-trained recognizer, testing recognition performance on the training, development and test sets. The results are shown Table 5.3.

Comparing the performance difference due to the complete language model (Tables 5.2 and 5.3), we see a small 3.2 percent improvement in the Letter Accuracy Rate. Thus, the obvious conclusion is that the performance of our two-stage system cannot be significantly improved by obtaining phonemic alignments for words that are expected to be used, but for which pronunciation is not available. This effect implies that after training on the full training set, the performance of the recognizer

Weight		Performance				
weightP2Ph	wPh2Sp	SR	DR	IR	LER	LAR
0.3	0	28.0	5.7	33.7	67.5	66.3
0.3	1.0	21.2	9.3	10.5	41.0	69.5

Table 5.4: Effect of the second ANGIE phoneme-to-spellneme stage on recognition performance.

begins to converge to its operational limit, suggesting that the phoneme-to-spellneme grammar generalizes well.

### 5.3.6 The Effect of the Phoneme-to-Spellneme ANGIE second-stage on Recognition Performance

A critical indicator of the success of our two-stage design would be some measure of the additional linguistic constraint supplied by our spellneme-driven second stage ANGIE “language model”. Specifically, we would like to know the proportion of recognition accuracy attributable to our second stage ANGIE FST - the focus of our design efforts described in Chapter 3. Conceivably, it could be the case that the training ANGIE probabilistic phoneme-to-spellneme grammars has no effect on recognition, and that our second stage is useful only as a lexicon for the usual ANGIE recognition, restricting the space of possible spellings, but not aiding in the hypothesis generation. This would support the unfortunate conclusion that no real benefit can be extracted from using the spellneme units as an additional linguistic constraint in the sound-to-letter recognition framework.

Fortunately, our experiments have conclusively established the importance and validity of the two-stage approach, as demonstrated by example results in Table 5.4. The removal of weight from the second stage ANGIE phoneme-to-spellneme model results in a 64.6 percent increase in the Letter Error Rate.

Evaluation set	SR	DR	IR	LER	LAR	Coverage
Full training for both ANGIE stages						
Development	7.3	4.4	2.1	13.8	88.3	72.0
Test	6.8	3.8	2.1	12.7	89.4	70.4
Complete second stage ANGIE language model						
Development	6.8	3.8	1.9	12.5	89.4	75.6
Test	6.3	3.4	1.9	11.6	90.3	72.6

Table 5.5: Forced phonetic recognition (FPR) experiments: the performance of the recognizer given correct input phonetics, for the parsed alignments in the development and test sets.

### 5.3.7 Forced Phonetic Recognition (FPR) Performance

Since most of the development effort in this thesis has been directed towards the construction of the two-stage ANGIE architecture centered around the spellname unit, the forced-phonetic recognition (FPR) experiment was a critical check of the viability of this approach. The goal of the experiment was to replace the output of the acoustic recognizer with ideal phonetic sequences for each word, and evaluate the resulting performance of the overall recognizer. A large increase in performance would indicate that most of the recognition error could be attributed to the SUMMIT acoustic recognition module, while the phone-to-spellname ANGIE composition is performing adequately. A related significant question was that of the utterance pronunciation coverage obtained by the ANGIE stages. Coverage in this context means the proportion of the “ideal” phonetic sequences parsed by the phone-to-spellname FST. Small coverage would imply that the ANGIE model/spellname grammar is sparse, and would undermine any conclusions drawn from the normal recognition experiments; indeed, the performance of a recognizer whose search space does not include the “correct” phonetic sequence for an utterance may be severely compromised.

As can be seen from the table, given sound phonetic sequences, the performance of the sound-to-letter recognizer improves dramatically. This seems to verify that the spellname multi-stage approach is feasible, and that the majority of spelling errors result from imperfect acoustic recognition.

### 5.3.8 Interpreting Forced Phonetic Recognition Results

The experiments with forced phonetic recognition provided us with valuable information on the performance of the multi-stage recognition framework. Specifically, on the basis of these experiments, we were able to divide the set of evaluated utterances into special subsets that reflected particular recognition patterns - for example, parse failures or perfect parses. Then, we evaluated these subsets separately in normal recognition mode, gaining insight on recognition performance under various conditions. These subsets and the main results of our findings are presented in Figure 5-3 and Table 5.6.

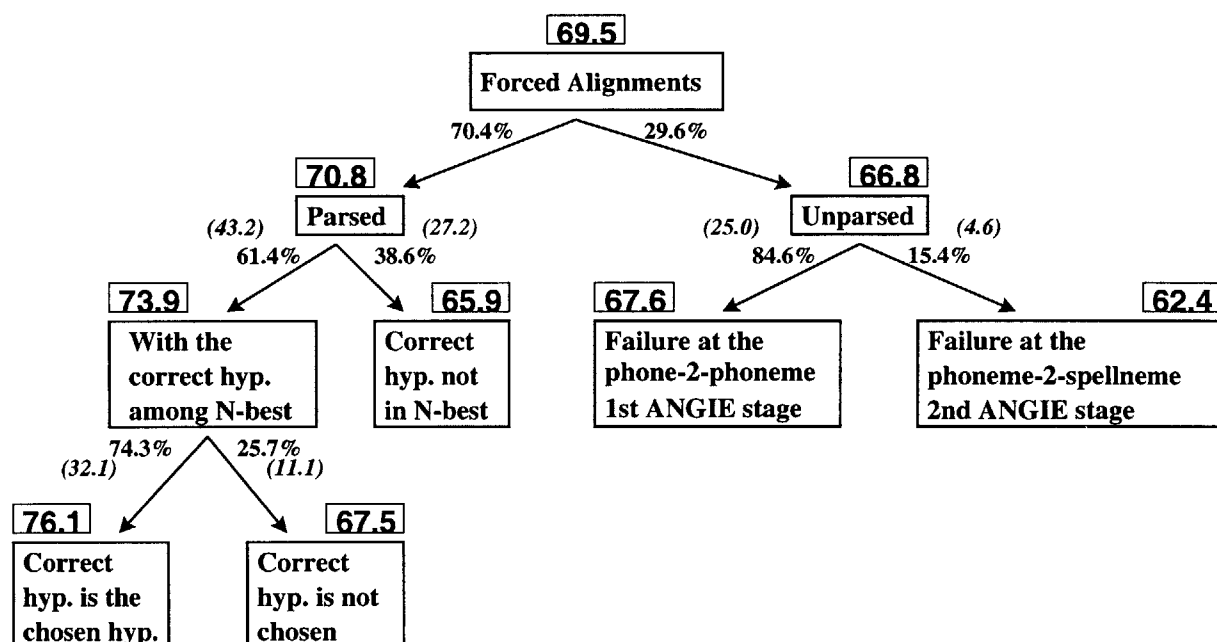


Figure 5-3: Decomposition of test set performance of a fully-trained (“fulltrain” for both ANGIE stages) recognizer according to forced phonetic recognition subsets. The boxed numbers are the LARs for individual subsets. Decomposition fractions for each subset are shown both as percentages of the parent sets and as percentages of the overall testing set (the later appear in parentheses).

At the top of the performance hierarchy is the test set of utterances. On the basis of the forced-phonetic recognition (FPR) experiments, the test utterances can be divided into the set of successful parses - those yielding at least one viable hypothesis (70.4 percent), and parse failures - instances in which FST composition failed because of coverage problems (29.6 percent). One would expect the FPR parse failures to

FPR set	SR	DR	IR	LER	LAR
Parsed	20.7	8.6	10.4	39.7	70.7
Failed on 1st stage	21.8	10.7	10.7	43.2	67.5
Failed on 2nd stage	25.5	12.1	10.4	48.0	62.4

Table 5.6: Normal recognition performance of principal FPR subsets on the test set.

perform poorly in normal recognition mode, for the reason that the “correct” phone sequence is absent from the two stage ANGIE-based phone-to-spellname mapping, for these utterances. Indeed, the performance of these utterances is lower, 66.8 LAR compared to 70.8 LAR for the successful FPR parses, but the difference is not striking. The relatively small performance gap between these two sets attests to sufficient richness of the phone-to-phoneme and phoneme-to-spellname grammars, which allows the formulation of a very close-matching hypothesis even in cases when the correct hypothesis is in some sense unattainable.

Another very interesting question is concerned with the relative performance of the two ANGIE stages in relation to parse failure. In this case, the FPR experiments enable us to identify the weak points in the proposed multi-stage framework. As Figure 5-3 indicates, almost 85 percent of the failed FPR parses are attributable to the first-stage ANGIE phone-to-phoneme module. Thus, we can reason that the multi-stage approach, and specifically the spellname-based ANGIE second stage that was the centerpiece of this thesis, has a significant positive effect on system performance (e.g. Section 5.3.6), at a comparably small coverage penalty.

Further insight on the value of N-best hypothesis lists can be extracted from the examination of the subdivisions of successful FPR parses. For a given  $N$ , two principal conditions are possible for a successful FPR parse of an utterance, corresponding to the presence or absence of the hypothesis with the correct spelling in the proposed N-best list. This simply means that the utterances whose forced phonetic sequences were “covered” by the recognizer can be divided into two sets: the first set consists of the utterances for which one of the top  $N$  recognition hypotheses produced the correct spelling (61.4 percent of all parsed utterances, as shown in Figure 5-3); the second

set contains the remaining parsed utterances (38.6 percent of the total). Further, the utterances for which the correct hypothesis is in the FPR N-best list, can be divided into a set of perfect parses, for which the correct hypothesis is the first entry in the N-best list, and the set of “imperfect” parses, in which the correct hypothesis is in the N-best list, but is not the top/selected choice. Thus, the “perfect” parses are those for which the entire words were recognized exactly, while the “imperfect” parses are those in which the correct spelling was contained in one of the top  $N$  recognition hypotheses, but not deemed the most likely by the recognizer. The data presented in Figure 5-3 were based on FPR experiments in which the top 50 hypotheses were considered. The first and logical result is that in almost two-thirds (61.4 percent) of the successful FPR parses, the correct answer is among at least the top 50 recognition hypotheses; further, utterances for which the correct answer is in the top 50 of the N-best list have a significantly higher LAR (73.9 vs. 65.9) than the remainder of the parsed set. Thus, the data verifies the expected correlation between the presence of the correct answers at the top of the N-best lists in FPR experiments and recognition accuracy, for a given utterance.

Finally, an interesting result stemming from detailed examination of FPR subsets is that roughly in three-fourths (74.3 percent) of the cases in which the correct answer is among the top 50 hypotheses, it is actually the best available lexical hypothesis, and the one chosen during FPR. Thus, the correct hypotheses, when present, are concentrated at the very top of their N-best lists - a finding that confirms the sharp initial slope of the cumulative N-best depth counts of Figure 5-2. In other words, this result indicates that if the correct spelling hypothesis for the entire utterance is considered “fairly likely” by the recognizer, then with a high probability it is actually the top lexical hypothesis.

### 5.3.9 Summary

While the major conclusions about the work performed in this thesis are the subject of Chapter 6, in this section we briefly recap the results of our experiments. The major preliminary task that we completed was the optimization of our multi-stage

recognizer - specifically, we discovered the optimal set of weights for the two ANGIE stages in composition (0.3 for phone-to-phoneme and 1.0 for phoneme-to-spellneme). Next, we evaluated the baseline performance of our recognizer, by training the optimal configuration on the full training set, and evaluating recognition accuracy on the test set. In the following step we compared test set performance with results from evaluation of the same configuration on the training and development sets, answering the question of how well the proposed recognition framework generalizes its training data to OOV utterances. In particular, we were reassured by the consistent performance of the recognizer on the test and development sets.

In addition to evaluating the recognizer’s generalization capabilities, we measured the effect of training data size on recognition performance, using recognition N-best lists to study the location of the correct hypothesis for variously trained recognizer configurations. The expected correlation between the amount of training and performance was discovered.

In the next experiment, we evaluated the possibility of enhancing the recognition power of our recognizer by augmenting the second-stage training lexicon with lexical knowledge from both test and development sets, motivating this step by the availability of phonemic data in real-life applications. However, we discovered that the resulting performance gains were not significant, a fact at least partly attributable to the completeness and generalization ability of the second-stage grammar.

Continuing with the assessment of the proposed multi-stage approach, we tested the influence of the spellneme-based ANGIE second stage on recognition performance, by comparing recognition accuracy of the optimal configuration with that in which no weight was assigned to the second stage. As expected, relaxation of the second-stage constraint severely curtailed recognition performance, reaffirming its importance in the multi-stage approach.

The final, and very significant set of experiments was based on the forced phonetic recognition model, in which we replaced the SUMMIT acoustics-to-phones stage with “correct” phone sequences, which we then fed into the upper two ANGIE stages. This procedure allowed us to assess the two ANGIE stages independently, and un-

cover the effect of poor acoustic recognition on overall performance. We found that under ideal acoustic recognition, the performance of the ANGIE FSTs was vastly improved, allowing us to conclude that much of the recognition error can be attributed to imperfect acoustic recognition, and that the ANGIE framework developed in this thesis performed its task adequately, obtaining 70-75 percent coverage on the correct phonetic transcriptions of the unseen words.

The second purpose for which the forced-phonetics experiments were very useful was the evaluation of comparative coverage power of the two ANGIE stages. Analysis of the unsuccessful ANGIE parses of the correct phone sequences allowed us to identify the phone-to-phoneme ANGIE stage as the principal culprit in coverage problems, raising hope for significant improvement in coverage, as explained in Chapter 6. Once again, the performance of the spellname-based second stage reaffirmed the validity of the multi-stage approach.



# Chapter 6

## Conclusions and Future work

The goal of the previous chapters was to present a multi-stage recognizer for the sound-to-letter problem. First, we provided background and motivation for the multi-stage architecture. Then, we introduced and developed a recognition framework based on a new sublexical unit, the spellname. Finally, we evaluated the performance of the proposed recognizer, exposing its weaknesses and strong points, and studying the role of individual stages in the overall recognition. The objective of this, the final chapter, is threefold: to underline the major results obtained in this thesis, to recount the lessons learned during design and development, and to connect the ideas developed in our work to ongoing research, casting the proposed recognizer into a larger vision of recognition systems.

### 6.1 Assessment of the Multi-Stage Approach

The focus of this thesis was on phonetic recognition, and specifically on the new phoneme-to-spellname ANGIE stage that was the critical piece of the multi-stage framework. Letter Error Rates of roughly 40 percent were achieved on unobserved words using normally trained recognizer configurations. The LER dropped to almost 10 percent when the acoustic recognition stage was replaced by “correct” phonetic sequences, suggesting that a large portion of recognition errors were due to incorrect acoustic recognition, and not the result of any inherent problems in the ANGIE-based

multi-stage approach.

The viability of the spellname-based second ANGIE stage, and its importance in the recognition framework was supported by three separate experiments. First, attempts to enhance the second-stage performance by adding a large number of additional training examples from the test data brought negligible improvement in performance, suggesting that the phoneme-to-spellname grammar succeeded in generalizing the sublexical mapping from the training utterances. The second relevant experiment measured the constraining effect of the second stage by evaluating recognition performance of a configuration that assigned zero weight to the phoneme-to-spellname FST during composition. It was found that removing the second-stage linguistic constraint increases the combined substitution, insertion and deletion error rate by 45 percent, implying that the proposed two-stage approach meets its objective of improving recognition accuracy through a sequence of tightly constraining stages. Lastly, forced phonetic recognition experiments showed that coverage problems were largely the consequence of sparse data problems in the first ANGIE phone-to-phoneme stage, while the spellname-based transducer registered failure in only 5 percent of the utterances. This is an encouraging outcome in that it should be feasible to train the first ANGIE stage on acoustic data from a wide range of sources in order to improve its coverage.

Thus, our experiments have clearly demonstrated the relative performance, generalization, and coverage benefits of the spellname-based multi-stage approach.

## 6.2 Future Improvements

The results obtained in the course of our work lead us to stipulate a series of improvements in the design of our recognizer. The first improvement is concerned with implementation observations, while the later two are directly motivated by the experiments described in the previous chapter, and specifically by the incomplete coverage attained by our recognizer.

### **6.2.1 Reusability of Recognition Modules**

Regarding our scheme of storing intermediate recognizer components in files with explicit names, we see this as a temporary replacement for an integrated recognition testing tool. Ideally, the various components and their inter-relationships are modelled explicitly as structures or classes, with member fields identifying the properties of their contents, their place in the hierarchy of recognizer construction, and the location of the physical files that store the data. In our experience, considerable effort has been spent on the administrative task of maintaining the component hierarchy, recording results, and making updates. The expended time calls for the integration of accounting tasks in large recognizers, and a uniform mechanism for managing various components, system parameters and results. Of course, such a system is beyond the scope of this thesis.

### **6.2.2 Extensive Phonetic Training**

As our forced experiments have shown, the majority of coverage problems in the ANGIE-based stages of our recognizer are caused by the sparse search space in the phone-to-phoneme grammar. Taking advantage of domain-independence in our sub-lexical models, we envision significantly more extensive phonetic training, including training on other domains, to enrich this space in order to reduce the number of coverage failures.

### **6.2.3 FST Back-off Paths**

Another technique that can be used to deal with incomplete column-bigram data makes use of FST back-off paths. Sparse data reduces the number of allowed transitions in an ANGIE-based transducer because certain column-bigram sequences remain unobserved, and thus not allowed in the trained FST. This may significantly impact the recognizer's coverage of OOV sequences. Essentially, the back-off paths method replaces the conditional probabilities of column-bigram pairs by a heuristic weight derived from the individual probabilities of occurrence for the involved column-bigrams.

Thus, for a slight performance penalty, one may gain significant improvement in recognition coverage.

### 6.3 Integration of the SLR into other Recognition Frameworks

We envision the proposed sound-to-letter recognizer as an important module in a powerful and flexible FST-based framework. In fact, we have identified a series of integration steps that have the potential to transform our recognizer into a powerful speech understanding tool. These steps are outlined in Figure 6-1.

One way in which our sound-to-letter module could be directly integrated into a larger recognition framework would be in the role of a post-processor for an OOV word recognizer. General OOV recognition is concerned with identifying the boundaries of out-of-vocabulary words in continuous speech, and is the subject of intense research [1]. The resulting tools are an ideal complement to an isolated-words recognizer such as ours, and their integration is a logical and elegant step in the overall speech recognition framework.

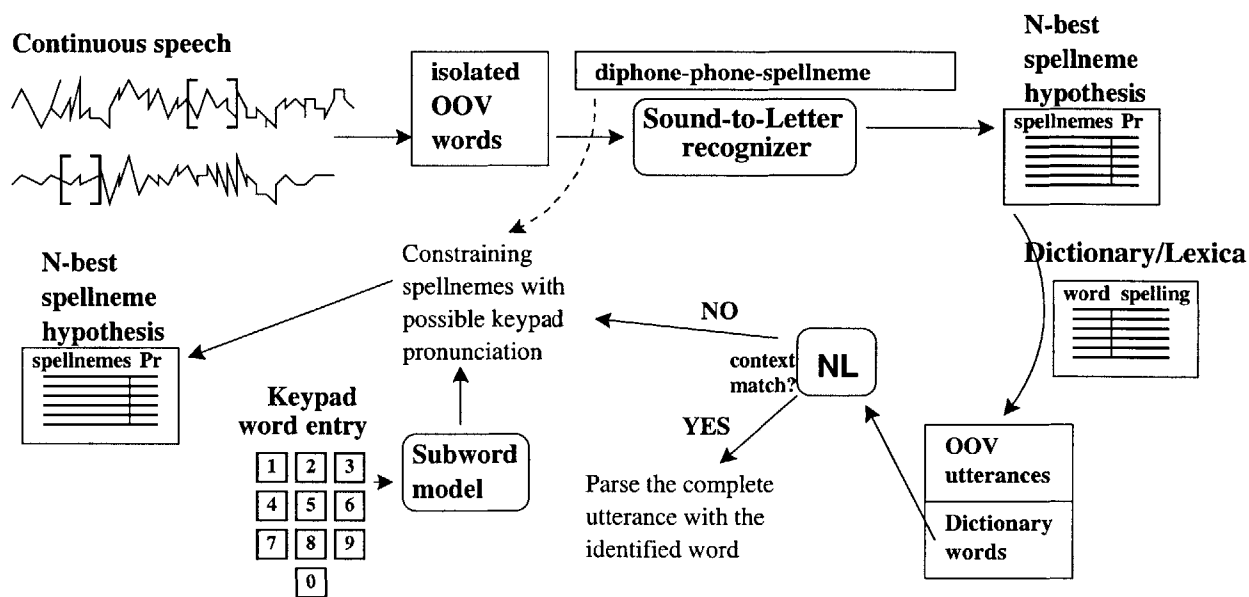


Figure 6-1: A roadmap of the integration of the proposed sound-to-letter module into a larger recognition framework.

### **6.3.1 Online Lexica/Dictionaries for Spelling Verification**

The first logical integration step deals with making sense of the recognition spelling hypotheses produced by our sound-to-letter recognizer. Our experiments have revealed that the task of producing perfect word spellings is a very difficult one, despite the fact that general letter accuracy results seem promising. Thus arises the need to map imperfect, but near-correct spellings to actual words. One way in which this can be accomplished, while preserving the OOV-recognition capability of the system, is to use large dictionaries and simple word morphology rules for this purpose. Given a spelling hypothesis, we can use substitution, insertion and deletion statistics to find the closest matching words in a dictionary. In this fashion, we can take an N-best list produced by the recognizer and obtain a list of the most likely words for each hypothesized spelling. At the same time, we also keep the top hypothesized spellings, since these may constitute OOV or out-of-dictionary (OOD) words that need to be defined.

### **6.3.2 Context-dependent Language and NL Models**

Ideally, the task of identifying isolated words will take place taking into account the surrounding context - both linguistic and semantic. This connection is particularly applicable to the result of extracting possible dictionary words from the N-best spelling hypotheses lists. For example, given the possible words “flight” and “bright”, the semantic constraint inherent in the sentence “I want to book a (???) to Boston” significantly reduces the difficulty of selecting the correct spelling of the unknown isolated word. Even if exact words cannot be identified by context, part-of-speech and morphology information may still prove very useful in pruning the search.

### **6.3.3 Solicited Word Spelling/Pronunciation**

A truly robust speech recognition system must have the capability of verifying its hypotheses through interaction with the user. The two possible modes of interaction are audio and keyboard-input, and both can be used to enhance recognition performance.

The simplest audio-based augmentation technique would ask the user to repeat certain words, and pool the scores from the various pronunciations in order to select the most likely hypothesis. Some kind of multiple-expert voting algorithm could be used for this purpose.

Another audio-based approach would entail asking the user to spell an isolated word letter-by-letter. One would hope that the task of identifying individual letters would be significantly less difficult, if not at all trivial. However, an obvious disadvantage of this approach is in the time required of the user, particularly if there are several OOV words in the utterance.

The solicited-input technique that is perhaps the most consistent with our overall vision is keypad entry of desired words. Indeed, whereas other audio-based methods will always be in some sense collinear, keypad-constrained spelling offers a powerful orthogonal constraint to audio information. In an ideal setting, the user would be able to use a keyboard to enter the complete spelling of a word; however, in line with the SLS telephone-based speech recognition framework, keypad entry of word spelling is a natural choice. The challenging aspect of including keypad information into the recognition scheme is the fact that keypad-based spelling is ambiguous, as a single keystroke may map to any one of three, or even four letters. Thus, it must be used as a constraint that prunes spelling derived from pronunciation. The blueprint for integrating keypad information with sound-to-letter rules has been set forth by the work of G.Chung and S.Seneff [8]. In their work, constraints imposed by keypad entries, a subword model, ANGIE column-bigram probabilities and phonological rules are integrated into a unified FST-based recognition framework. A possible procedure for performing this integration with our sound-to-letter recognizer may consist of the following steps:

1. A keypad FST ( $K$ ) is created by mapping a sequence of  $n$  keypad strokes into a transducer with  $n + 1$  states, such that every consecutive pair of states is joined by 3 or 4 transitions. This FST gives the space of possible spellings for the keypad entry.

2. The keypad FST is composed with a subword bigram ( $B$ ), which constrains word spellings based on probable morph sequences. In order to map the letters to morphs, an intermediate FST ( $L$ ) is required. This step produces an FST ( $G$ ), where  $G = K \circ L \circ B$ .
3. Next, we project the morphs to their letters, and prune  $G$ , to obtain an FST that maps letters allowed by the keypad entry to probable spellings.
4. In the penultimate step, the input and output sequences of  $G$  are reversed, and the result is composed with a simple FST that maps spellnemes to letters (this FST can be derived directly from the spelling information contained in the spellnemes. Thus, we obtain an FST  $X$  that maps spellnemes into the space of letter sequences allowed by the keypad entry.
5. Finally, we compose the transducer from the previous step with the diphone-to-spellneme FST developed in this thesis, to obtain a final keypad-constrained sound-to-letter recognizer.

## 6.4 Summary

Overall, we believe that the results obtained in this thesis demonstrate the scientific and technological viability of the spellneme-based multi-stage approach to sound-to-letter recognition. A rich set of applications motivates a wide range of ideas on how to integrate the proposed recognizer into current and future research, with the ultimate goal of producing a definitive recognition framework. We have outlined some of these ideas in this chapter, and are looking forward to future work in this promising area of speech understanding.



# Bibliography

- [1] I. Bazzi and J. Glass. Learning units for domain-independent out-of-vocabulary word modelling. In *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, September 2001.
- [2] G. Chung. A three-stage solution for flexible vocabulary speech understanding. In *Proc. International Conference on Spoken Language Processing*, Beijing, China, October 2000.
- [3] G. Chung. *Towards Multi-Domain Speech Understanding with Flexible and Dynamic Vocabulary*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, June 2001.
- [4] J. Chang J. Glass and M. McCandless. A probabilistic framework for feature-based speech recognition. In *Proc. International Conference on Spoken Language Processing*, volume 4, pages 2277–2280, Philadelphia, PA, October 1996.
- [5] K. Livescu and J. Glass. Segment-based recognition on the phonebook task: Initial results and observations on duration modeling. In *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, September 2001.
- [6] F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*, pages 431–453, Cambridge, MA, 1997. MIT Press.
- [7] P. Price. Evaluation of spoken language systems: the atis domain. In *Proc. DARPA Speech and Natural Language Workshop*, pages 91–95, Philadelphia, PA, 1990.

- [8] G. Chung S. Seneff. Integrating speech with keypad input for automatic entry of spelling and pronunciation of new words. In *TBA*, 2002.
- [9] R. Lau S. Seneff and H. Meng. Angie: A new framework for speech analysis based on morpho-phonological modelling. In *Proc. International Conference on Spoken Language Processing*, Philadelphia, PA, October 1996.
- [10] A. Park T.J. Hazen, I.L. Hetherington. Fst-based recognition techniques for multi-lingual and multi-domain spontaneous speech. In *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, September 2001.
- [11] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington. JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):100–112, January 2000.
- [12] V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goodine, and D. Goddeau. PEGASUS: A spoken dialogue interface for on-line air travel planning. In *Proc. International Symposium on Spoken Dialogue Systems*, Tokyo, Japan, November 1993.

3231-91