Towards Multi-Domain Speech Understanding with Flexible and Dynamic Vocabulary

by

Grace Chung

S.M., Massachusetts Institute of Technology (1997)
B. Eng., University of New South Wales (1995)
B. Sc., University of New South Wales (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author ....
Department of Electrical Engineering and Computer Science
June, 2001

Certified by..
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by ....
Arthur Smith
Chairman, Departmental Committee on Graduate Students

Towards Multi-Domain Speech Understanding with Flexible and Dynamic Vocabulary

by

Grace Chung

Submitted to the Department of Electrical Engineering and Computer Science
on June, 2001, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Abstract

In developing telephone-based conversational systems, we foresee future systems capable of supporting multiple domains and flexible vocabulary. Users can pursue several topics of interest within a single telephone call, and the system is able to switch transparently among domains within a single dialog. This system is able to detect the presence of any out-of-vocabulary (OOV) words, and automatically hypothesizes each of their pronunciation, spelling and meaning. These can be confirmed with the user and the new words are subsequently incorporated into the recognizer lexicon for future use.

This thesis will describe our work towards realizing such a vision, using a multi-stage architecture. Our work is focused on organizing the application of linguistic constraints in order to accommodate multiple domain topics and dynamic vocabulary at the spoken input. The philosophy is to exclusively apply below word-level linguistic knowledge at the initial stage. Such knowledge is domain-independent and general to all of the English language. Hence, this is broad enough to support any unknown words that may appear at the input, as well as input from several topic domains. At the same time, the initial pass narrows the search space for the next stage, where domain-specific knowledge that resides at the word-level or above is applied. In the second stage, we envision several parallel recognizers, each with higher order language models tailored specifically to its domain. A final decision algorithm selects a final hypothesis from the set of parallel recognizers.

Part of our contribution is the development of a novel first stage which attempts to maximize linguistic constraints, using only below word-level information. The goals are to prevent sequences of unknown words from being pruned away prematurely while maintaining performance on in-vocabulary items, as well as reducing the search space for later stages. Our solution coordinates the application of various subword level knowledge sources. The recognizer lexicon is implemented with an inventory of linguistically motivated units called *morphs*, which are syllables augmented with spelling and word position. This first stage is designed to output a phonetic network so that we are not committed to the initial hypotheses. This adds robustness, as later stages can propose words directly from phones.

To maximize performance on the first stage, much of our focus has centered on the integration of a set of hierarchical sublexical models into this first pass. To do this, we utilize the ANGIE framework which supports a trainable context-free grammar, and is designed to acquire subword-level and phonological information statistically. Its models can generalize knowledge about word structure, learned from in-vocabulary data, to previously unseen words. We explore methods for collapsing the ANGIE models into a finite-state transducer (FST) representation which enables these complex models to be efficiently integrated into recognition. The ANGIE-FST needs to encapsulate the hierarchical knowledge of ANGIE and replicate ANGIE's ability to support previously unobserved phonetic sequences. The result is the column bigram ANGIE-FST which captures ANGIE probability parse scores on the FST arcs, and treats a parse tree as a sequence of vertical columns.

During the course of our work, we conceived of a new phoneme-level inventory of symbols called *letter-phonemes* which codify both spelling and pronunciation. These are a set of grapheme units annotated with pronunciations, as well as linguistic properties such as stress. They are embedded in the ANGIE parse tree at the pre-terminal layer, so that the ANGIE probability models encode spelling along with other sublexical phenomena. We have found that augmenting with letter-phonemes leads to a reduction in perplexity. Additionally, a hypothesized spelling for an unknown word can be retrieved directly from the parse tree during recognition simply by extracting the letter-phonemes, proposed at the pre-terminal layer.

The final implementation of stage one comprises an FST that incorporates ANGIE models with grapheme information, together with constraints from a lexicon of *automatically derived* morph and sub-morph units. These new units are optimized from an iterative procedure employing the column bigram FST and letter-phoneme units. They are altered in terms of spelling and syllabification from the original morph lexicon. The result is a more compact final FST stemming from a smaller ANGIE grammar that has improved probability likelihoods.

Also in our thesis work, we designed a second stage to search over a phonetic network employing the original ANGIE parser to process the phonetic sequences and identify possible unknown words. The control strategy couples ANGIE parsing with word-level models, and probabilistic natural language (NL) models are an option. In the later part of our work, we experiment with a third stage which shifts the application of NL to the final stage on a small word network produced during the second pass.

A final set of recognition experiments is performed on sentences containing unknown city names, from the JUPITER weather information domain. We evaluate on recognition and understanding compared with a baseline system with no OOV handling capability. Using a three-stage system, we produced up to 67% relative improvement in understanding performance. We also demonstrate a preliminary ability to instantaneously extract spelling hypotheses of the unknown word at recognition time.

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Acknowledgments

I would like to express my deepest gratitude to my advisor Stephanie Seneff. It has been almost six years since I first arrived at the Spoken Language Systems Group, and I have been extremely fortunate to have worked with Stephanie throughout this time. It has been great fun, to say the least. Stephanie has been particularly inspiring in her creativity and genius. I have thoroughly enjoyed and appreciated being the recipient of her wealth of visionary ideas. She is an extremely understanding advisor. What's more, her unrelenting energy and enthusiasm are truly amazing.

At SLS, it has been a very positive and nurturing environment to work in. I most appreciate the opportunity to spend these years at MIT alongside so many talented people. It was a rare and wonderful experience to have the chance to work on a list of intriguing and challenging problems, and to have at hand the resources, stimulus and support to find creative solutions to them.

I would like to thank Victor Zue for his guidance and vision. He has done countless things to make life as a graduate student here more productive and stimulating. I owe to Victor plenty of valuable suggestions regarding my research, and many opportunities to speak about my work at conferences throughout the years. His leadership has made SLS an outstanding place to be.

I would like to offer special thanks to Lee Hetherington for his assistance in my understanding of finite-state transducers. He is singly responsible for the underlying implementation of the FSTs. He has been an invaluable resource, being the resident expert on the FST algorithms.

I would also like to thank Ken Stevens on my thesis committee for useful feedback on my work. He has been most valuable for offering different perspectives on this work. I am also grateful to all the members of SLS, including all the dedicated staff members. And I am sad to be leaving my cool officemates, Xiaolong Mou and Ed Filisko.

Finally, of course, special acknowledgements for all my family and friends, who have made my life in the United States most colorful. Never a dull moment in between long hours in front of the computer.

Grace Chung

June, 2001

Contents

1 Background **19**

1.1 Introduction 19

1.2 Conceptual Vision 21

1.3 Thesis Goals 22

1.4 Experimental Domain 24

1.5 Previous Research 24

1.5.1 Introduction 24

1.5.2 The Unknown Word Problem 25

1.5.3 Integration of Linguistic Knowledge with Speech Recognition 28

1.5.4 The Use of Syllable Knowledge in Speech Recognition 30

1.6 Overview 33

2 The Linguistic Model Design **37**

2.1 Introduction 37

2.2 The Linguistic Hierarchy 38

2.3 A Multi-Stage Architecture 40

2.3.1 A Conceptual Solution 40

2.4 Linguistic Modeling Issues of Stage One 42

2.4.1 The Morph Unit 42

2.4.2 The Phonetic Network 43

2.4.3 Lexical Generation 45

2.5 A Comment on Stage Two 45

3 The Representation of Hierarchical Sublexical Knowledge **47**

3.1 Overview 47

3.2 An Introduction to ANGIE 47

3.3 Motivations of ANGIE 48

3.3.1 Sublexical Modeling 48

3.3.2 Theoretical Background 50

3.4 The Framework of ANGIE 51

3.4.1 Introduction 51

3.4.2 The Grammar 53

3.4.3 The Lexicon 56

3.4.4 The Dynamic Parse Mechanism and Probability Models 57

3.4.5 Discussion 59

3.5 Past Applications of ANGIE 62

3.5.1 Letter-to-Sound 62

3.5.2 Duration Modeling 62

3.5.3 Subword Modeling and Flexible Vocabulary 63

3.6 Finite-State Transducers 65

3.7 ANGIE and the FST Representation 68

3.7.1 A Tree-like Graph Structure 69

3.7.2 An FST Structure Based on ANGIE Columns 71

3.7.3 Additional Comments 73

3.8 Incorporating Grapheme Information 73

3.9 Final Remarks 75

4 A Preliminary Two-Stage System **77**

4.1 Overview 77

4.2 Motivation 77

4.3 System Architecture 78

4.4 Stage One 80

4.4.1 The Lexicon: Words versus Morphs 80

4.4.2 Acoustic Modeling 82

4.4.3 Language Modeling and Search 82

4.5 Network Creation 83

4.6 Stage Two 84

4.6.1 The ANGIE Parse Mechanism 84

4.6.2 The TINA Framework 84

4.6.3 The Integrated ANGIE-TINA System 86

4.7 Recognition Experiments 89

4.7.1 Experimental Method 89

4.7.2 Understanding Evaluation 90

4.7.3 Results and Analysis 91

4.8 Discussion and Summary 94

5 A Finite-State Transducer Based System **97**

5.1 Overview 97

5.2 Motivation 97

5.3 System Architecture 98

5.4 Stage One: The FST-Based System 99

5.4.1 The ANGIE Grammar 101

5.4.2 The ANGIE FST 102

5.4.3 FST Generation Procedure 104

5.4.4 Additional Comments 106

5.5 Stage Two 106

5.6 Experiments 107

5.6.1 Method 107

5.6.2 Results and Analysis 108

5.7 Discussion 111

5.8 Final Remarks 112

6 Improvements for the First Stage **115**

6.1 Overview 115

6.2 Design Considerations 115

6.3 A Spelling-Based ANGIE Grammar 117

6.3.1 Introduction 117

6.3.2 Letter-Phonemes 118

6.4 The Column-Bigram FST 125

6.4.1 Introduction 125

6.4.2 Column Bigram Structure 126

6.4.3 Coverage and Smoothing 128

6.4.4 Discussion 130

6.5 A Revised Two-Stage Architecture 132

6.5.1 Stage One 132

6.5.2 Stage Two 135

6.6 Recognition Experiments 135

6.7 Final Remarks 138

7 Automatic Lexical Generation 141

7.1 Overview 141

7.2 Motivation 141

7.3 Lexical Space Optimization 142

7.4 The Iterative Procedure 144

7.5 Results of Iteration 146

7.5.1 Perplexity Measurements 148

7.5.2 The Novel Lexicon 148

7.6 Final Remarks 152

8 Unknown Word Experiments 155

8.1 Introduction 155

8.2 Three-Stage System 156

8.2.1 Stage One 157

8.2.2 Stage Two: The ANGIE-Based Search 157

8.2.3 Stage Three: TINA Parsing 160

8.3 Experiments 161

8.3.1 Training 161

8.3.2 Recognition 161

8.3.3 Results and Analysis 162

8.3.4 Spelling Extraction 165

8.3.5 Running Times 167

8.4 Final Remarks 168

9 Conclusions 171

9.1 Summary of Contributions . . . 171

9.1.1 A Multi-Stage Approach . . . 172

9.1.2 Sublexical Modeling and Finite-State Transducers . . . 173

9.1.3 Novel Symbolic Representations involving Graphemes . . . 174

Letter-Phonemes . . . 175

The Morph Unit . . . 175

Novel Subword Units . . . 176

9.1.4 Demonstrating Flexible Vocabulary . . . 176

9.2 Future Directions . . . 177

9.2.1 A Multi-Domain System . . . 177

9.2.2 Flexible Vocabulary . . . 178

Combining with Confidence Scoring or a Rejection Mechanism . 178

Learning an Unknown Word . . . 179

9.2.3 Continuing Research in ANGIE Integration . . . 180

A Glossary 181

B A Guideline on Notation 189

B.1 Morphs . . . 189

B.2 Letter-Phonemes . . . 189

C Example Context-free Rules for Letter-Phonemes 191

C.1 Low Level Rules . . . 191

C.2 High Level Rules . . . 196

Bibliography 201

List of Figures

2-1 *A Proposed Linguistic Hierarchy for Speech[69].* 39

2-2 *A Multi-Stage Multi-Domain Speech Understanding System.* 41

3-1 *Sample Parse Tree for the Phrase “I’m interested...” Below the word level are layers representing morphology, syllabification, phonemics and phonetics. At morphology layer are nodes for: function word (*FCN*), stressed root (*SROOT*), unstressed root (*UROOT*), derivational suffix (*DSUF*) and inflectional suffix (*ISUF*). At the syllabification layer are nodes for: nucleus and coda in a function word (*FNUC *and* FCODA*); vowel nuclei that are (1) stressed and lax (*NUCLAX+*), (2) unstressed (*NUC*), and (3) in a derivational suffix (*DNUC*); stressed and unstressed coda (*CODA *and* UCODA*); a past tense *PAST*. The phonemic layer contains pseudo-phonemic units with contextual markers such as “+” for lexical stress and word context such as “_I” and suffix context such as “*ed” for the past tense suffix. The terminal layer consists of phonetic symbols. A /-n/ marks the deletion of the parent phoneme /t/, tied to the left context /n/.* 52

3-2 *Tabular Schematic of a Parse Tree. The phrase is I’m interested ..* 58

3-3 *Tabular Schematic of ANGIE Parse Trees. The words represented are days and place.* 60

3-4 *Tabular Schematic of ANGIE Parse Trees. The word depicted is plays.* 61

3-5 *Schematic of a Left-to-Right Branching FST. Words with common phonetic sequences advancing from the left, will share common arcs.* 70

3-6 *Schematic of a Right-to-Left Branching FST. Outputs are emitted from the left, and arcs are successively merged together from left to the right.* 70

3-7 *Schematic of One Type of FST Supporting OOV Words. There are alternative pathways supporting OOV sequences in addition to in-vocabulary data.* . . 72

4-1 *Block Diagram of the Preliminary Two-Stage System.* 79

4-2 *Illustration of the* ANGIE-TINA *Integration Strategy. See Section 4.6.3 for detailed explanation.* 87

5-1 *Block Diagram of FST-Based Two-Stage System.* 99

5-2 *Morph-Based ANGIE Grammar Parse Tree. This represents the morph bos+ for the word boston. The morphology layer simply categorizes the morph bos+ as a stressed syllable.* 101

5-3 *A Simplified FST for the Morph "bos+" in Boston. Labels on the arcs denote the inputs and outputs, separated by a ":". Situated in the self-looping arc, a "#" is a special symbol denoting a morph boundary, as the arc returns to begin another phonetic sequence belonging to the next morph. The pause marker, "<pau>", marks the beginning and end of a sentence.* 102

5-4 *Step by Step Summary of FST Generation Procedure.* 104

6-1 *Schematic of an FST for the Word Days. The input and output labels, given at the arcs, are delineated by a ":". The phonetic realization is given by /d ey z/, and the letter-phoneme baseform is given by /d! ay_l+ s*pl/. The "isuff" label is emitted indicating the inflectional suffix at the column associated with the plural and the /s*pl/ letter-phoneme.* 128

6-2 *Schematic of the Column Bigram Structure. A path for the word "days" has been captured in this FST. C_1 is the state that corresponds with a column with phone terminal /d/ and C_2 is the state that corresponds with a column with phone terminal /ey/. The state marked with "*" is the back-off state connecting from an onset column to other rhyme columns.* 129

6-3 *Tabular Schematic of ANGIE Parse Trees. The words represented are days and place.* 131

8-1 *Block diagram of Three-Stage System.* 158

8-2 *Bar Graph of Word and Understanding Error Rates.* 163

List of Tables

3-1 *Example of a Two-Tier Lexicon of ANGIE. See text as well as Appendix B for explanation of the meanings of the diacritics.* 56

4-1 Set of Compound Function Words in the JUPITER Domain. 81

4-2 *Typical Examples of Utterances and their Key-Value Pairs Used in the Understanding Evaluation.* 90

4-3 *Recognition and Understanding Performance. Errors are given for systems described in Section 4.7.1. Systems 1 and 2 represent single-stage baselines, and Systems 3 and 4 represent two-stage systems with word lexicons at the first stage.* 92

4-4 *Recognition and Understanding Performance for Two-Stage Systems. These use morph lexicons instead of word lexicons at the first stage.* 93

4-5 *Comparison of Morph Error Rates for Selected Systems. See text for explanation.* 94

5-1 *Word and Understanding Error Rates for the Development Set.* 109

5-2 *Morph Error Rates for the Development Set. The first line, the SUMMIT top 1, is the top scoring sentence from the single-stage baseline. The second line gives the morph output of the first-stage recognizer that is used in the two-stage architecture of Systems 5 and 6. A morph trigram is used in the first stage. The third line is the second stage output of the two-stage system of System 5, where ANGIE is used without TINA.* 109

5-3 *Word and Understanding Error Rates for the Test Set.* 110

6-1 *A List of Letter-Phoneme Categories. These are arranged in the major phonemic categories. Stressed vowels are appended with a "+". "_l" appends a long vowel. "_x" appends a lax vowel. "_uns" appends an unstressed rhyme. "!" appends a consonant in onset position. "_fcn" appends vowels in function words. See text for further explanation and see Appendix B for a list of meanings of annotations.* 119

6-2 *An Excerpt from the Letter-Phoneme Morph Lexicon. Example morphs are the stressed roots can+, cane+ and rain+ and the derivational suffix, -tion.* 122

6-3 *Example of a Morph and its Decomposition into the Onset and Rhyme. Illustrated is the stressed root "prince+." An appended "=" annotates an onset and a prepended "=" annotates a rhyme.* 132

6-4 *Complete List of Syllable Onsets Decomposed from Stressed Roots. (An appending "=" denotes the stressed syllable onset.)* 133

6-5 *Complete List of Syllable Rhymes Decomposed from Stressed Roots. (A prepending "=" denotes the stressed rhyme.)* 134

6-6 *A Comparison of FST sizes, using the Column Bigram as the ANGIE-FST. FSTs with and without smoothing are included. P represents the ANGIE-FST with ANGIE derived probabilities as the arc weights. U represents a final composed FST embedded all language model constraints.* 136

6-7 *Performance on In-Vocabulary* JUPITER *data. Word (WER) and understanding (UER) error rates (%) are quoted for an independent test set of 1806 utterances, comparing two single-stage* SUMMIT *baseline systems with bigram and trigram language models and the two-stage system described in this chapter.*137

7-1 *Size of FSTs with Different ANGIE Grammars.* 147

7-2 *Probability Measurements of Column Bigram ANGIE-FSTs. Various ANGIE grammars are compared. See text for explanations.* 149

7-3 *Examples of Modified Spellings for Words and their Morph Decompositions. Some words and their corresponding morphs before and after the application of our iterative procedure are shown.* 149

7-4 *Example of Re-syllabification. The syllabification of the word Antarctica before and after the application of our iterative procedure is shown.* 150

7-5 *Modification of Word Boundary Affiliations for Consonants. Examples of words and their corresponding morphs before and after the application of our iterative procedure are shown.* 150

7-6 *Modification of Word Boundary Modifications for Morphs. Examples of words and their corresponding morphs before and after the application of our iterative procedure are shown.* 151

7-7 *Examples of Foot-like Compound Units Derived from Iterative Procedure.* . 151

7-8 *More Examples of Sentences with Novel Words Compared With Their Original Orthographies.* 152

8-1 *List of Concepts and Example Values for Understanding Evaluation.* 162

8-2 *Comparison of Parse Failures During Evaluation for 4 Systems. Results given for the 425-utterance test set. A marked drop in the number of failures is observed for the systems that employ* TINA *(System 2 and 3).* 165

8-3 *Breakdown of Word Error Rates (WER) for baseline system and three experimental systems on a 425-utterance test set with unknown city names.* . . . 165

8-4 *Breakdown of Understanding Error Rates (UER) for baseline system and three experimental systems on a 425-utterance test set with unknown city names.* 165

8-5 *Error rates for Letter Recognition of Unknown Cities. There were 164 cities in total.* 166

8-6 *Top 20 Confusions for the Letters Hypotheses.* 166

8-7 *Examples of Letter-Spelling Hypotheses from the Two-Stage Recognizer. Spellings were extracted from the letter-phonemes at the pre-terminal layer of the* ANGIE *parse tree.* 168

8-8 *Average Computation Time Per Utterance. The test set averages 7.9 words in length.* 169

Chapter 1

Background

1.1 Introduction

In recent years, large numbers of spoken dialog systems[1] are being developed around the world, both as research prototypes and commercial applications [107, 26]. These are increasingly employed as human-computer interfaces used for retrieving information, conducting transactions and performing various interactive problem-solving tasks. In most cases, systems function within highly restricted domains. Typically, a user interacts with a system in order to seek some information from local databases or from the Web. Some examples are flight scheduling [111], air travel information [85], train timetable information [22], weather information [110] and direction finding [27]. The extent to which a system takes an active role during conversation can vary. While some are machine-directed, that is, very restrictive, other dialog systems use a mixed-initiative approach, permitting greater flexibility for a user to specify demands for a task.

The emergence of the above applications can be directly attributed to advancement in speech recognition and language understanding technology, and their integration in recent years. Meanwhile, there remain many challenging research problems in order to meet with increasing demands of these applications. In the future, spoken dialog systems will need to access information from a broad variety of sources and services, such as on-line databases, and to operate on not one but across several restricted domains. A system must then allow users to switch automatically among several domains in a seamless fashion in order for users

[1] We use the terms spoken dialog system, spoken language system and conversational interface interchangeably.

to perform multiple unrelated tasks in a single call. One core problem arising from such a system is the ability to handle unknown words. In real-world applications, particularly with mixed-initiative dialog systems, the presence of out-of-vocabulary (OOV) items is inevitable. Previous research has shown this to be true, even for very large dictionary vocabularies [38]. OOV items may arise from queries that stretch beyond the range of information known to the system, such as an unknown city name; they may be queries that are entirely out of scope (e.g. a traffic information query in a weather information domain); or they may simply occur as artifacts of spontaneous speech such as word fragments and disfluencies. In fact, it is impossible to anticipate all the potential words, the partial words or even the nature of queries that could be posed by the many potential users. Moreover, as the information being accessed is likely to change frequently and unpredictably, the vocabulary on the part of the system is bound to constantly evolve in response to the dynamic information content. This would apply for domains that include vast numbers of proper names such as surnames, restaurant names or place names. Thus, this scenario calls for future systems to routinely extend vocabularies on-the-fly without the need for a human to explicitly enter the new words, with their pronunciations, retrain the models and restart the system.

With conventional systems, vocabularies tend to be closed or fixed. When unknown words emerge at the spoken input, these systems can either reject the utterance altogether or alternatively, propose at the unknown word an erroneous hypothesis with an acoustically similar profile. Frequently, one unknown word engenders several subsequent errors so that the sentence is completely misunderstood. This often happens because the error at the unknown word perturbs the probability scores from the language model, ultimately favoring additional errors on surrounding words. Yet the system continues the discourse, unaware of any errors committed, whereas immediate failure or backtracking would have been a more appropriate route. The result is user frustration and confusion, eliminating any hope for error recovery. The user remains unaware that the query resided outside the boundaries of system knowledge, and the system is unable to inform the user which portion of the query was out of scope for the topic domain.

A more ideal scenario is an architecture that supports a flexible and dynamically extensible vocabulary. Here, when an input sentence contains a previously unseen word, the system detects its presence, provides appropriate feedback, and attempts to infer the word's meaning, spelling as well as acoustic and phonological properties. After confirming these properties with the user, the system then increments its vocabulary with the new word which can be used at any time during future conversations.

1.2 Conceptual Vision

Conceptually, we envision a telephone-based[2] spoken language interface which is capable of supporting multiple domains simultaneously, while a dynamic and flexible vocabulary resides within each domain. This enables users to pursue several topics of interest within a single telephone call, and, in response, the system switches transparently from one domain to another without explicit notification.

A vocabulary that is flexible and dynamic would mean that the system can cope with unknown words both at the user voice input as well as from the information source. An ideal scenario, for example, could involve a user query containing a previously unseen word. Here, we assume that, for this domain, it is not possible to anticipate and include all words that can be spoken into the speech recognizer. An example would be a general directory assistance application where the number of place names and surnames that could be queried is very large and constantly changes over time, while directories themselves may not be up to date. The system is able to detect the presence of the OOV and, following this, uses acoustic, phonological and linguistic knowledge to hypothesize a set of phonemic sequences for pronunciation and a set of letter sequences for plausible spellings. Additionally, from the dialog context, the system deduces the semantic class, such as a city name. The system then proceeds to browse its information databases for the closest match to its hypotheses and to respond to the original user query. Furthermore, in the ideal scenario, the system may choose to verify with the user on the proposed pronunciation and spelling. Having successfully arrived at a representation for the new word, the system immediately incorporates it into the lexicon.

This scenario can have several variants. For example, the system can learn new words via a spell-mode where the user enters the spelling and pronunciation of the word verbally at the input. Alternatively, as the system retrieves information from a database in response

[2] Our work is not confined solely to telephone-based systems, and the proposed architecture can be adopted for all spoken dialog systems. However, effective dynamic vocabulary acquisition plays a more important role in a spoken language interface environment that is impoverished of alternative modalities such as a keyboard input and visual display.

to a user query, such as a list of cities in California, the system may notice certain cities absent from its own lexicon and decide to add those incrementally. It does so with the knowledge that all new words are city names, and specifically, those located in California. This information is noted for use in higher order language models, including natural language (NL) models. Because built-in flexibility in the system obviates the need to retrain lexical and language models, large numbers of words can be easily added in this way.

1.3 Thesis Goals

This thesis addresses the design issues surrounding the realization of our conceptual vision. One major challenge lies in conceiving a modeling framework where, upon encountering unknown words, the system detects them as such and appropriately deduces their linguistic properties. These functionalities must be performed without compromising accuracy on understanding in-vocabulary or in-domain data. Our main point of interest is the organization of the linguistic model, which needs to offer both flexibility and tight constraint. The philosophy of our strategy centers on the existence of a linguistic hierarchy that spans from the phone to the discourse level. And, in order to exploit this, we formulate our language models around these multi-level constraints which we intend to apply in a tightly coupled manner. But this inevitably causes the size of our search space to balloon and demands us to manage the search process more intelligently.

In light of these issues, we envision a solution that comprises the following elements. First of all, the combination of many disparate sources of linguistic knowledge necessitates the use of a multi-stage paradigm. In particular, we advocate the application of low-level and domain-independent linguistic constraints as early as possible in order to prune the search space. That is, the first stage exclusively utilizes low-level knowledge of generic English[3]. Meanwhile, the use of word-level and higher order language models is delayed until later stages. These later stages will also support multiple domains. This can be done for instance by using several recognizers in parallel, each with its own set of domain-specific models. The obvious alternate solution for this is to implement a single monolithic system that accommodates for all possible sentences that can be spoken, with coverage spanning

[3] This thesis will only address a system based on English. It is beyond our scope to evaluate our system on other languages, although we postulate that all of our methodologies can be viably applied towards other languages.

over all the topic domains. It may be possible that a future system can accomplish this. However, it will have to grapple with an enormous search space and complexity issues in terms of time and space. When dealing with several possible disjoint conversational topics, that is the spoken queries are largely unrelated, we suspect that multiple separate recognizers would provide better constraint, and ultimately afford better performance. We will elaborate on these concepts in Chapter 2. Much of our research then focuses on the problem of capturing and organizing the knowledge derived from below that of the word level. This is directed towards building an effective first-stage recognizer. Our success will hinge largely upon the combination of two recent innovative advances: (1) ANGIE, a computational model designed to capture word substructure from the bottom-up, lending itself to possibly extrapolate OOV information from in-vocabulary knowledge, and (2) finite-state transducer (FST) technology, which has gained popular acceptance in recognition systems for efficiency and potential computational savings. The concept of merging these two technologies will be discussed further in Chapter 3.

During the course of this thesis, we will evaluate the feasibility of each aspect of our ideas through a series of experiments with prototypical systems. In the beginning, we will show that the integration of complex models in a two-stage configuration can be effective, and can yield high performance compared with a state-of-the-art single-stage system. A major contribution will be the transformation of the ANGIE framework into an FST representation, allowing this complex set of sublexical models to integrate with the recognizer search. We will introduce and evaluate different methods for constructing ANGIE-based FSTs. Further on, we will show that grapheme[4] information can be valuable both for enhancing low-level linguistic constraint as well as allowing automatic access to spelling hypotheses. Moreover, we develop a stage-one system that both incorporates grapheme information and employs automatically generated lexical units with optimized sublexical probabilities. We demonstrate that these ideas make potentially significant progress towards a multi-domain flexible vocabulary system that handles OOV words intelligently.

[4] A grapheme is a character used in writing. It may have varying realizations or allographs. For example, the grapheme "a" subsumes the variants or allographs "A" and "a." See a glossary in Appendix A for a detailed definition.

1.4 Experimental Domain

Ideally, we envision a first stage that utilizes information drawn from large corpora of general English, to maintain domain independence, while later stages support several distinct recognizers developed for different domains. But the scope of this thesis is limited to investigating the feasibility of our designs. Therefore, all of the experiments are confined to the JUPITER domain.

JUPITER [28, 110] is a telephone-only mixed-initiative spoken dialog system for weather information for more than 500 cities worldwide. The weather information is obtained from on-line sources on the Web, and is updated several times daily. The most recent version contains up to 650 cities, 166 countries and about 2000 words in the vocabulary.

One reason for choosing this domain is that it is a real-world application in which the spoken input, characteristic of highly spontaneous speech, frequently contains OOV words and out-of-domain queries, responsible for major performance degradation in the state-of-the-art system. A first attempt towards exploring issues in flexible vocabulary would be to examine the OOV set. The JUPITER domain is very suitable due to (1) its richness in proper nouns such as place names and (2) its highly restrictive task goal so that a semantic category for an unknown word is relatively predictable, for example a city name. Our results will be compared with a baseline system, similar to one that is available for public use.

The training of the first stage models on a large general corpus and the implementation of several restricted topic domains are reserved for future work. It is also beyond the scope of this thesis to address the issue of switching between several domains.

1.5 Previous Research

1.5.1 Introduction

Over the next subsections, we review some of the background material to our work, including previous approaches to various aspects of the research problem and past research that has inspired and contributed to our own. While previous research has not directly addressed the design of a system such as ours, researchers have touched upon some of the individual issues that are relevant in this thesis. Here, we summarize some work performed with regard to handling unknown vocabulary in speech recognition, strategies for integrating linguistic knowledge with the recognition search, and techniques for representing syllable knowledge in recognition. Each of these topics has been part of our consideration; our own work has largely entailed integrating disparate linguistic knowledge at multiple stages, particularly syllable information, towards the goal of handling unknown words.

1.5.2 The Unknown Word Problem

Hetherington's work [38] established that even in systems with a very large dictionary vocabulary (exceeding 100,000 words), the OOV rate can exceed 1%. Assuming a sentence length of twenty words, this translates to one or more OOV words in nearly one in five sentences. His work revealed that each OOV word causes on average 1.5 recognition errors; that is, the presence of an unknown word leads to additional errors for surrounding in-vocabulary words. When the recognizer is forced to substitute the unknown word with an acoustically similar in-vocabulary word, thereby introducing an error, the language models cannot be relied upon to predict correctly the surrounding words, hence engendering more errors. It can be concluded that the presence of unknown words significantly impairs overall understanding accuracy for spoken dialog systems. In fact, in the JUPITER domain [110], word error rates for out-of-domain sentences escalate to more than double the in-vocabulary error. These sentences range from legitimate queries containing unknown words such as city names to out-of-domain queries, as well as utterances with spontaneous speech artifacts such as word fragments.

At present, most conventional recognizers use only a few rudimentary methods to circumvent this problem. For instance, some simply attempt to maximize coverage and minimize the OOV rate by increasing the number of entries in the lexicon. In reality, this is not practical. Not only are many unknown words actually proper names, not listed in large dictionaries, but also a portion constitutes word fragments which cannot be covered by a finite dictionary. Many systems resort to utterance rejection techniques [6, 7], whereby the objective is to classify the recognized output as correct or misrecognized, via some confidence measures [7, 91]. While this technology is commonly used to provide user feedback of any difficulties that a system may be experiencing with the spoken input, it falls short of offering opportunity for the system to identify or learn a new word. Others [37, 6] attempt to locate a possible unknown word by essentially operating a word and a phone recognizer in parallel, and comparing their respective scores. Classification techniques are used to determine a threshold for rejecting a word as an unknown. While this obviates the need for an explicit unknown word model, its reliability is limited, and performance improvements have been small.

Other work has primarily modeled OOV items via a generic word or garbage model as an extension to the in-vocabulary lexicon. Usually, this acoustic model is a concatenation of phonemes with some constraints imposed on the sequence. This then competes with other in-vocabulary items, and is hypothesized if its *a posteriori* probability exceeds that of the others. The work of Asadi et al. [3, 4] first used this technique to model unknown words on the Resource Management corpus, an artificial domain, and experimented with various strategies for transcribing the new word with the aid of a letter-to-sound system, subsequently adding the new word to the vocabulary. This early work did not impose any word structure knowledge to the phoneme sequence in the garbage model, although there were some attempts to model the pronunciation rules of the unknown word. Through the years, others have embraced and further investigated this approach [100], but have achieved little in terms of bridging the performance gap between sentences with and without unknown words, and even less in terms of transcribing and learning a new word.

Some have attempted to incorporate syllable knowledge in hopes of exploiting phonotactic constraints. Kemp [52] proposed the use of a syllable-based method for the acoustic modeling of new words for a German corpus. He built a generic syllable model for unknown words but met with limited success. Similar to our ideas, De Mori [18] envisioned a vocabulary-independent recognizer, and considered basic lexical units such as syllables that encompassed more constraining information on the search space than pure phonemes. He proposed generating a network in a first stage consisting of "pseudo-syllable" acoustic units. A second pass decoded the network into word hypotheses. A pilot experiment in detecting new words from data extracted from the Wall Street Journal task was conducted. He demonstrated some success by detecting half the OOV words from a twenty sentence test set. Recently, Klakow [57] demonstrated more success by augmenting the lexicon with automatically generated word fragments. Like previous work, these fillers were variable-length phoneme sequences but several hundreds of them were generated and trained on some artificial data prior to recognition time. Their goal was to reduce the performance degradation incurred by OOV items, and also possibly propose a phonetic transcription for the OOV region. This filler approach was reminiscent of that used in word-spotting tasks, and was also undertaken by Meliani [68]. Meliani had also found marginally better performance in using syllable-based fillers.

Thus far, we have surveyed research in acoustic modeling of unknown words. As is important in our system design, there remains the aspect of handling the language model upon an unknown word hypothesis. Jelinek et al. [46] studied the problem of incorporating new words into a statistical n-gram model. Such a model requires a large amount of training data to estimate all the word n-tuple frequencies. Their solution was to assign a new word to word classes based on context. Most others have followed along similar lines. However, the rigidity of the n-gram modeling paradigm and the generally poor estimation of language model parameters have contributed to mediocre performance on the handling of unknown words. Some have attempted to overcome these barriers. Damnati et al. [17] discussed an island-driven parser which extracts a conceptual class for the unknown word before determining the word class. Boros et al. [5] addressed issues in dealing with OOV items in the semantic processing and the dialog manager. As akin with our objectives, Boros was concerned with improving usability, and building a more cooperative spoken dialog system, upon encountering OOV data.

Little previous work has addressed issues of enrolling new vocabulary words into a recognizer and eliciting phonetic baseforms automatically. Researchers [86] have reported simple techniques such as generating the baseform using acoustics alone and expanding that with phonological rules. In order to increase robustness, systems may require users to enter multiple samples of the new word, offering a system a better chance of proposing the correct phonetic transcription [34].

Some work which has experimented with acquiring new word spellings directly from acoustics was that of Alleva and Lee [2]. This was a first study of an acoustic-to-spelling transcriber, aimed at automatically acquiring both the phonetic and orthographic transcriptions for new words. It is of particular interest to us because our work shares the same objective of deducing spellings from unknown words. Their novelty stemmed from using actual letter spellings in their context-dependent Hidden Markov Model (HMM) inventory instead of phoneme units, such that the spelling of unknown words could be directly extracted from the acoustic model representation. Although this strategy was highly innovative, the experiments carried out were mostly preliminary, and results seemed unpromising. This drove the authors to recommend the alternative of using a separate sound-to-letter module for deriving spellings instead. In [92], Schillo et al. attempted a grapheme-based recognizer whereby graphemes replaced phonemes in the acoustic models. They reported grapheme and word accuracies on a German corpus of 10,000 city names.

1.5.3 Integration of Linguistic Knowledge with Speech Recognition

It has been argued by some [69, 96, 107] that human speech understanding involves integrating numerous knowledge sources such as discourse, semantics, syntactics and prosodics. These knowledge sources can be organized into a linguistic hierarchy. When these constraints are applied in concert in a tightly coupled fashion, they can provide mutual feedback to guide and aid the search process during recognition. In particular, it is important to pay careful attention to the interaction between linguistic components and the speech recognition component. While the NL component serves to extract a meaning representation for an utterance, it also needs to tolerate errorful recognition hypotheses and artifacts of spontaneous speech such as false starts and repairs. However, it should also constrain the recognition search to penalize unlikely hypotheses based on their meaning and grammatical structure. Thus, there is potential to steer the acoustically driven recognition search based on factors such as syntax and semantics. The main obstacle here is that limitations in computational resources preclude the simultaneous deployment of rich but cumbersome higher order language models. In many instances, we are forced to apply each constraint sequentially.

In the past, the most popular integration strategy has been the N-best interface [9, 73, 108, 39], where the NL component acts as a post-processor, filtering on whole sentence hypotheses from the recognizer. Analyzing one sentence at a time, the NL component will stop at the first sentence when a meaning is successfully extracted. The NL component can help identify misrecognized sentences, that is, act as a rejection mechanism, by discarding sentences that do not parse grammatically. Furthermore, NL systems that can output probability scores [39] can combine these acoustic and linguistic scores to re-order the N-best list, and choose the most likely hypothesis, based on the various combined knowledge sources. However, this process, in its entirely feed-forward nature, is suboptimal, in that the NL system cannot interact with and provide feedback to the recognition search. In an effective integration strategy, the NL component could potentially eliminate portions of the search space much earlier using syntactic and semantic analyses, while promoting linguistically meaningful candidates.

More recently, word graphs have become a popular alternative to N-best lists as the interface to a language processing module, following the first-pass recognition stage [53]. Word graphs are both easily generated [75, 76], and a more compact representation than the N-best list representation. They are now widely employed for post-processing by a variety of knowledge sources, from higher order n-gram language models to dialog [102] and prosody modules [58]. The flexibility allows multiple simultaneous information sources as well as multiple-stage architectures. In many instances, researchers [32, 98, 35, 47] have developed technology to parse sentence hypotheses directly from word graphs, applying semantic and syntactic constraints. They have proven that understanding accuracy can benefit when the NL knowledge, aided by trained probability scores, promotes those hypotheses which are more meaningful, and this can be better achieved when the integration enables the search to probe deeper via a word graph. Some [35, 47] have also discovered that applying higher order constraints using a word graph can speed up overall computation time.

Nonetheless, much has yet to be explored in the way of tighter coupling between NL and recognition, beyond the word graph paradigm. After all, the word graph technique inflicts hard decisions by irretrievably removing portions of the original search space. In principle, it still appears more desirable to incorporate some NL analysis directly within the recognition search, with minimal prior pruning. And for a spoken dialog system, it is even more convenient if a full meaning representation can be generated at the same time. But clearly, the extraction of a full or partial meaning representation early in the recognition process presents difficulties, such as requiring a prohibitive amount of computation. This then has spurred researchers to seek alternative strategies [30, 72, 101, 24] with varying degrees of complexity.

In [30], Goddeau formulated a probabilistic language model based on an LR parsing algorithm, and applied this to a speech understanding task. The shift-reduce parser was integrated with an A^* search, and yielded an increase in the number of utterances correctly understood. With only modest performance results, this work suffered some main drawbacks: (1) it was unable to generate a full meaning representation and (2) upon the event of a parse failure, it resorted to the use of the word n-gram model. In Ward's work [101], the goal was to exploit longer span language constraints in the decoding process by using methods that are sufficiently robust in order to cope with spontaneous speech. He described a collection of recursive transition networks (RTN) each representing semantic fragments. The A^* decoder would prefer but not require hypothesized sequences that comply with the RTN grammar. Although there was no stochastic component, a significant understanding improvement was attained. In Jelinek's recent work [45], the Structured Language Model (SLM) used a hierarchical structure to extract meaningful information from word sequences, and was intended to outperform an n-gram model because of the presence of longer distance language constraints.

Much of the abovementioned work, in actuality, meant enhancing the language model with limited syntactic/semantic information. By contrast, our work strives for a complete NL analysis together with a meaning representation, generated during the integrated search phase. In Chapter 4, we will introduce our hierarchical NL processor, TINA, and describe how it has been tightly incorporated within the acoustically driven search.

1.5.4 The Use of Syllable Knowledge in Speech Recognition

For years, phonologists have come to understand the importance of the syllable as a linguistic unit and its indispensable role in phonotactic constraints [5]. For example, Randolph [87] completed a comprehensive study of the acoustic realizations of data, and concluded empirically that the syllable was instrumental in explaining realizations, particularly in stop consonants. More recently, many speech researchers [78] have argued the inadequacy of the phoneme as a representational unit for recognition because of its apparent dependence on higher-level structure. Disappointing gains in phone pronunciation modeling for conversational speech [88] have further suggested that a flat model relying exclusively on the phoneme is inherently flawed. Many are beginning to espouse more linguistically-motivated approaches, mostly from an acoustic modeling point of view. By contrast, our ideas will attempt to capture in the language models those constraints that occur within the framework of the syllable.

Fujimura [20] first proposed using the syllable for speech recognition. Since then, there has been growing interest for using subword units larger than phones but smaller than words in acoustic modeling. This interest is motivated by mounting evidence that syllable-level timings are perceptually more meaningful, more attuned with the function of the human

[5] We will discuss this point in greater detail later as part of the theoretical motivations of ANGIE.

auditory system, and therefore less susceptible to speaking rate effects than phonemes [33]. This would also imply that syllable-level representations are more robust in noisy environments. In conventional recognizers, boundaries of phoneme-level units are often difficult to elicit, in spite of the existence of powerful context-sensitive models. On the other hand, syllable units with their longer time windows are better equipped to capture dynamic coarticulatory effects and pronunciation variations. In fact, syllable units are claimed to form natural representations for capturing suprasegmental, prosodic and metrical information as well as phonotactic information [106, 42].

Current research is beginning to embrace the use of syllable-level acoustic units. In general, the inventory of subword units is derived via a variety of methods, from automatic data-driven approaches to more empirical or knowledge-driven approaches. The main challenge lies with the difficulty of syllabification[6], and arriving at an appropriate set of units. Hu [41] merged phones together into larger "syllable-like" units at locations where segment boundaries are difficult to discern. Her experiments, conducted on a small-vocabulary recognizer, yielded results comparable to those of a recognizer using phone-like units. Jones [49, 50] implemented a syllable recognizer for a 1300-word read speech task. His results affirmed that comparable word recognition accuracy can be achieved using syllable units. Yet, the lack of training data, given that the inventory of syllables was much greater than phones, posed a problem. To alleviate this, Jones varied the number of Gaussian mixtures in his models according to the amount of data available.

Similar work was also undertaken by Hausenstein [36]. Hausenstein applies syllables as the unit for classification in a hybrid neural network HMM recognition system on various digit corpora. He lengthened his window of analysis to capture syllable-level acoustic events but kept the feature set the same.

Alternatively, one could use a hybrid system with units ranging from phones to words. This idea was adopted by Shukat-Talamazzini in [99], in which the subword units, referred to as context-freezing units (CFUs), were derived from a hierarchical decomposition of a word representation. This exploited structural knowledge more effectively than triphone units. Similar work was carried forth by Pfau et al. [84] where "macro-demisyllables" (MDS) were automatically created. The procedure started with demisyllables and successively concate-

[6] Ambisyllabicity and determination of syllable boundaries remain subjects of debate among phonologists [93, 51].

nated them using a global optimization criterion at each iteration. Other innovative work includes that of Kirchoff, in using speech-production motivated phonetic features in syllable models [56]. Here, acoustic syllable models were described by a host of speech articulatory features. Similar goals were pursued by King et al. [54] who envisioned entirely abandoning current phone-based HMMs, and ultimately using syllable models with parameters that explicitly describe trajectories for syllable phonetic features.

So far, the most success has been attained when syllable information is not a substitution for other models but used in conjunction with them. The work of Jones et al. [48] used a small set of syllable-sized HMMs to model syllable effects, resulting in 23% reduction in word error rate on the TIMIT corpus. They applied syllable knowledge to the N-best output of a recognizer that used conventional knowledge sources. The syllable HMMs utilized stress information and prosodic features. Wu et al. [105, 104] used both syllable- and phone-scale information by combining HMM recognizers each with different feature representations. This work employed the modulation spectrogram, a new feature extraction method that is motivated by recent findings in speech perception and psychoacoustics, highlighting the role of a portion of the frequency spectrum (2-16Hz) most pertinent to syllable timings [33]. Her results reported that syllable-based systems failed to outperform phone-based systems under clean conditions. When she combined hypotheses of the phone and syllable recognizers at the *syllable* level during decoding, performance gains were achieved, and this was most pronounced under reverberant conditions.

These trends suggest a potential for the syllable to play an even greater role in recognition in the future. Yet developments have tended towards the acoustic modeling perspective where this knowledge is encoded, by and large, in a flat and unstructured probabilistic framework so that little control can be exercised. While it is certain that syllable-level information in acoustic modeling offers benefits, our work promotes an alternative but not exclusive path. In the first place, the ANGIE framework is an engineering solution which aims to capture formally the hierarchical nature of sublexical information. Central to this is the encapsulation of phonotactic constraints which occur with respect to the syllable unit. This is independent of the choice of acoustic units, relying entirely on the linguistic models to represent syllable-based phenomena. Secondly, as will be further explicated in Chapter 2, our first-stage recognizer employs syllable-sized lexical units defined as *morphs*. These also partially encode syllable-based knowledge among other linguistic factors such as spelling and lexical stress.

1.6 Overview

Now that the reader is acquainted with both the major ideas of this thesis and some related previous research, the following outlines the remainder of the thesis. The next two chapters serve to lay out the foundation of our work. We hope the reader will glean from these an understanding for the origin of the composite of ideas we have assembled. The later chapters provide full expositions of our algorithms, the properties of our symbolic representations, our experimental methods and results. The chapters are delineated as given below.

- **Chapter 2: The Linguistic Model Design**

  We will explore in greater detail (1) issues concerning the design of a multi-stage architecture, (2) the organization of hierarchical linguistic models within this configuration, and (3) considerations for selecting linguistic constraints that contribute towards a low-level and domain-independent first stage.

- **Chapter 3: The Representation of Hierarchical Sublexical Knowledge**

  This is an introduction to our hierarchical sublexical model, ANGIE. This chapter will trace the development of ANGIE and present its theoretical underpinnings. It will also explicate the probability framework and outline the benefits of an ANGIE-based system. We then pose some of the engineering problems that were encountered in the past, and introduce the possibility of solutions utilizing FST technology. After providing an overview of properties of FSTs and their current applications, we will argue for the potential merits for adopting an FST representation. In particular, we introduce the concept of (1) translating the ANGIE framework into an FST and (2) augmenting the ANGIE-based FST with grapheme information.

- **Chapter 4: A Preliminary Two-Stage System**

  Our first recognition experiment is presented. An initial two-stage system which integrates ANGIE with TINA, the NL module, is developed. We explore the feasibility of two basic design elements: (1) the use of syllable knowledge in the first stage and (2) the use of a phonetic network as the interface between stages.

- **Chapter 5: A Finite-State Transducer Based System**

We show further progress towards building a domain-independent first stage, by enhancing the current syllable-based first-stage with ANGIE-based sublexical knowledge. This is accomplished by an FST-based implementation of the first-stage recognizer. The new configuration proves to further benefit performance, and can be computed in near-real-time. We will discuss the advantages of using the new FST paradigm but emphasize the inadequacies that remain for this architecture when addressing the flexible vocabulary problem.

- **Chapter 6: Improvements for the First Stage**

  We next present some breakthroughs in designing our initial stage which are directed towards solving the problem of supporting OOV without sacrificing performance. We are principly interested in improving our first stage by introducing greater generality yet optimizing on low-level linguistic constraints. This prevents phone sequences of unknown words from being pruned away during the search. We elucidate on a novel method for constructing an ANGIE-based FST called the *Column Bigram*. And secondly, a new device for simultaneously modeling phonological and letter-to-sound phenomena within the ANGIE framework is introduced. We detail the characteristics of a set of *Letter-Phoneme* units, that are invented to encode spelling, pronunciation and other contextual properties. And we describe the new improved two-stage system combining these elements.

- **Chapter 7: Automatic Lexical Generation**

  This chapter is devoted to the idea of generating a novel subword lexicon. We will describe the considerations which have driven us to this point. This is followed by an explication of an iterative procedure for generatively deriving novel syllable-sized lexical units for the first-stage recognizer. We present the final results upon implementing the iterations to convergence and give an analysis on this novel set of units.

- **Chapter 8: Unknown Word Experiments**

  A summary of the final system will be given. We evaluate two and three-stage versions of our design by testing on JUPITER sentences with unknown city names. We also attempt to extract spelling hypotheses for these unknown words instantaneously. We shall discuss the implications of the success of these experiments.

- **Chapter 9: Conclusions**

  Finally, we revisit the findings of this thesis, and provide some suggestions for future experiments.

A glossary of our definitions for the terminology used in this thesis has been included in Appendix A to facilitate the reader in resolving any uncertainties that may arise. And in Appendix B, we have provided explanations to the annotations used in denoting linguistic units throughout this thesis.

Chapter 2

The Linguistic Model Design

2.1 Introduction

In the previous chapter, we defined our vision for the capabilities of a flexible vocabulary system in Section 1.2, and in Section 1.3, we broadly introduced the scope of our work that will further us towards this goal. Here we address an issue that is central to our objectives: the selection and organization of various sources of linguistic constraints. In selecting the types of constraints, it would be necessary to determine the symbolic representations that the modeling framework must employ.

The major challenge is a reliable method for the detection and recognition of OOV items. This is difficult to accomplish without causing degradation in recognition performance on in-vocabulary data. As it is often used, a catch-all or generic OOV word model can be deficient in terms of constraint for capturing an unknown word but can also disrupt recognition on the surrounding adjacent portions of the sentence with known words. Moreover, current systems lack the mechanisms to deduce the linguistic properties of a detected unknown word.

In fact, the main dilemma we are faced with stems from an inherent trade-off between a need for increased flexibility versus tighter linguistic constraint. Higher-level domain-specific knowledge can provide the necessary constraint. However, when an OOV word occurs, the system's models will naturally favor the hypothesis of an in-vocabulary item with an acoustically similar profile. The reason is that zero or very low probabilities are allocated to unknown words whose phonetic sequences are generally previously unobserved. But incorporating more flexibility will enable a system to recognize or license sequences of phones that (1) do not occur in words existing in the pre-determined dictionary and (2) have not been instantiated in the training corpus. If the acoustic and linguistic models incorporate sufficient generality, they would offer probability support despite the lack of training realizations. However, there now emerges a conflict wherein large amounts of flexibility necessarily leads to a relaxation of language constraints. But general recognition demands the application of more tightly constrained language models as early as possible to prune away unlikely hypotheses. Hence, increased flexibility often compromises the performance on in-vocabulary and in-domain input.

In the following, it is shown that once various sources of linguistic information are arranged in the form of a hierarchy, these can be applied successively within a multi-stage architecture. With this in mind, we propose one possible conceptual solution in Section 2.3.1 that could implement a flexible vocabulary system. This serves as a starting point for developing our ideas throughout this thesis. Of primary importance is the design of the first-stage linguistic framework. We proceed to discuss some of our key ideas for assembling the initial stage, and conclude with some comments on the second stage.

2.2 The Linguistic Hierarchy

In Meng's thesis [69], it was argued that speech and language processes can be arranged under a hierarchy which spans from the acoustic level up to the paralinguistic and discourse level. Meng conceived of a hierarchical scheme, illustrated in Figure 2-1. The hierarchy takes into account phenomena such as phonological processes, syllabification, prosodics and semantics. As this structure is said to reflect the human communication process, it can also be seen as an array of knowledge sources available to a speech recognizer. It is crucial to take into consideration the elements of this hierarchy and their possible interactions when encoding linguistic knowledge.

Let us consider low-level or sublexical constraints, that is, those which reside below the word level and which capture patterns within general word substructure. These are valuable because they express phenomena relevant to all of the English language[1]. And in

[1] By all of the English language, we refer to the entire set of English words, many of which are borrowed from foreign languages. One example is the word "schlerosis." Our below word-level models are expected to learn from the training data the characteristics of all English words including those influenced by other languages. We note that a language such as German or Mandarin Chinese with fewer foreign borrowings and more regular subword structures could perform even better with our modeling framework.

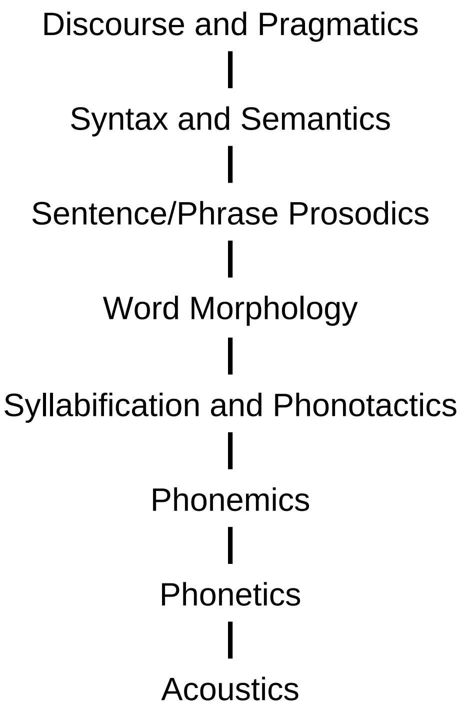

Figure 2-1: *A Proposed Linguistic Hierarchy for Speech[69].*

consequence, they are not exclusively applicable to the current topic domain. The intrinsic domain-independence is useful because in principle, such models would not discriminate between in-vocabulary data and OOV data. In fact, this model could even support recognition of partial words. In our work, we have regarded low-level constraints as the key to linguistic support for OOV items. On the other hand, high-level constraints such as those describing syntax and semantics could be used to determine where an unknown word is more likely to occur.

Given the abovementioned scheme, we envision an architecture which fulfills the following:

- When multiple constraints in the linguistic hierarchy are applied, they are tightly coupled with one another and with the acoustics-based search. This promotes mutual interaction by positive feedback while guiding the exploration of the search space.
- Constraints are applied in a configuration where hypotheses, corresponding with previously unobserved data, are not prematurely pruned away.
- Allowing the system to propose unknown words and sequences that are unsupported by training data does not impair recognition accuracy on input which falls within the

current topic domain.

- All the above is accomplished without placing intractable demands on computational resources.

The final point above is one of major challenge because the combination of models that encode the rich hierarchy of language represents an explosively large search space[2]. At this point, this issue precludes their combined integration within a single-stage recognition engine.

2.3 A Multi-Stage Architecture

Our proposal is to use a multi-stage system architecture. As remarked in Section 1.5.3, integrating multiple knowledge sources has been possible in the past using N-best lists and word graphs which represent a pruned-down search space constructed from a first-pass recognizer.

In our case, we propose specifically to use a multiple pass approach that applies knowledge sources from the linguistic hierarchy successively from the bottom layer upwards. Thus, each pass utilizes information from higher levels in the hierarchy, and reduces the overall search space by some amount. This philosophy implies that only low-level linguistic information is used at first while the application of high-level constraints is delayed. It also implies the separation of domain-independent from domain-dependent components. We illustrate this general strategy further by presenting a conceptual multi-stage architecture in the next section.

2.3.1 A Conceptual Solution

Consider the multi-stage architecture depicted in Figure 2-2. The initial stage is a core recognition engine which only utilizes low-level domain-independent models. These models draw upon general acoustic and linguistic knowledge that are representative of all of the English language. In particular, the linguistic component codifies properties such as phonotactic, phonological, syllable and morphological information. The probability models can be trained on several large and general corpora. On the one hand, we believe that sublexical

[2] This point is demonstrated in our past experience in working with integrating our hierarchical language models ANGIE and TINA. We will discuss more about each of these later on.

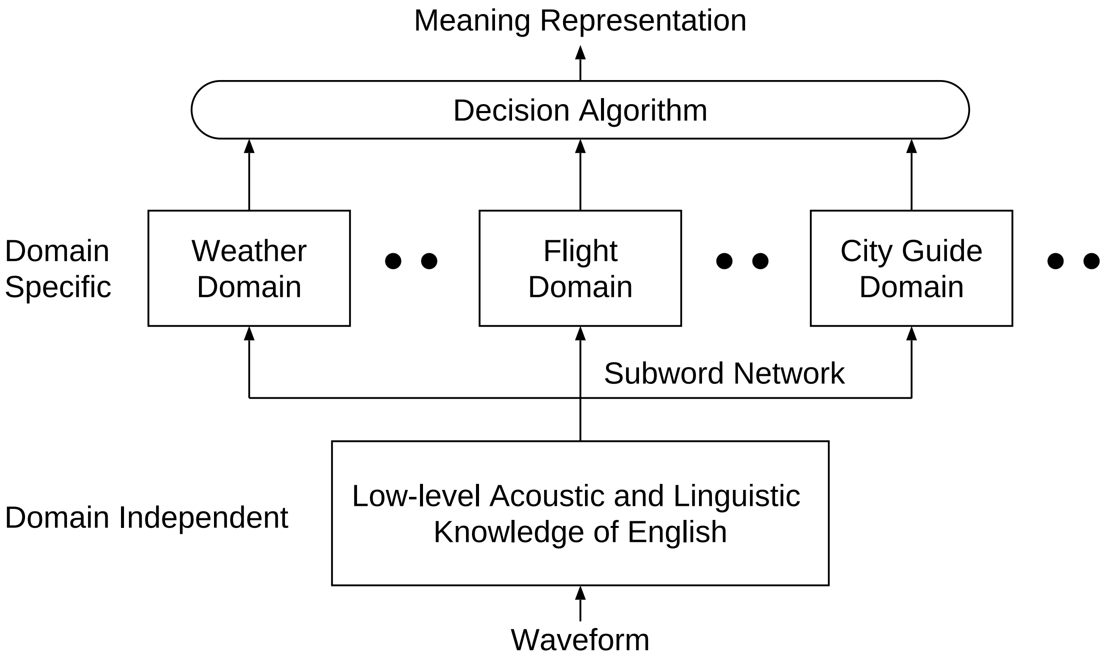

Figure 2-2: *A Multi-Stage Multi-Domain Speech Understanding System.*

knowledge can constitute a powerful language model to facilitate recognition. But it is also a general source of constraint whereby OOV but plausible English word constructions can be given probability support. One way to realize this, for instance, is to construct a lexicon for a large number of subword-sized units. These subwords may be syllables, although the exact lexicon design is subject to further investigation. Doing so frees us from the confines of a fixed domain-dependent word lexicon, and could ultimately cover much of the English vocabulary, including even partial words. We will further elaborate on these ideas in the next sections.

The output of the first stage is a subword lattice which is then processed by a suite of domain-dependent speech understanding modules, operating in parallel. With a smaller search space, each of these second-stage modules can apply an array of higher-level linguistic information without an explosion in computational requirements. These richly embedded linguistic models will be domain-specific, accounting for word-level knowledge as well as natural language processing and dialog context. The key to this architecture is an effective domain-independent first stage which eliminates significant portions of the search space. Consequently, the later stage can enlist powerful long-distance models and apply them in an integrated manner. The final decision for the best meaning representation is mediated by a top-level decision algorithm. This allows the user at the input to switch among domains at any time without prior notice.

2.4 Linguistic Modeling Issues of Stage One

The previous sections have established the role of a domain-independent low-level first stage. Much of this thesis will investigate the multiple issues arising from the linguistic design of this first stage. It is certain that the lower echelons of the linguistic hierarchy have much to offer in terms of constraint, but it is still indeterminate how this knowledge can be captured. We grapple with the following issues:

- What lexical units best enforce constraints that are sufficiently low-level and can generalize across the English language?
- What symbols can be incorporated into our lexical representation to optimize on contextual knowledge?

The ANGIE framework will be critical in providing a mechanism for describing and predicting sublexical phenomena. This will be detailed in Chapter 3. But for now, we introduce and motivate some novel concepts that underlie the development of the first stage. These will operate in conjunction with ANGIE models. The engineering effort in implementing these ideas and the experiments which verify their feasibility will be presented in detail in the later chapters.

2.4.1 The Morph Unit

In considering the design of the recognizer lexicon, our investigations have led us to the inception of a linguistically motivated syllable-like unit which we refer to as the *morph*[3]. As intimated earlier, many phonotactic constraints occur within the context of the syllable unit, and using only local phonemic context would not exploit the internal structures embedded within English words. If we erected a recognizer using only phonemic level information, this would satisfy our criterion for generality but longer distance contextual information would be lost. Without higher level information to disambiguate between confusable phonemic

[3] We refer to the *morph* unit exclusively given by our definition. This may differ from the usage of this term by others in the literature. Please refer to the glossary, given in Appendix A, for more definitions of terminology used in this thesis.

hypotheses, the result is likely to cause a large number of competing hypotheses. Subsequently, an unacceptable compromise on performance, particularly for in-vocabulary words, would occur. By contrast, using word-level units soon becomes unwieldy. No word-level recognizer can be truly domain-independent, and cover all novel constructions and partial word possibilities. At the syllable level, the inventory is much smaller and tractable.

Yet, as mentioned in Section 1.5.4, the syllable remains an elusive unit of representation. It is a natural means of representation in Asian languages, but is still difficult to capture in English because of issues such as ambisyllabicity. Here, we believe that a syllable-level lexicon would be most appropriate given our dual goals for achieving generality and constraint. Our engineering solution is to formulate the notion of a morph unit. We define this as a syllable-sized unit augmented with additional linguistic properties such as spelling, stress, morphology and position. Like the syllable, a morph can be decomposed into the constituents such as nucleus, onset and coda. In fact, these morphs are represented by the spelling of the unit, and variations in letter casings encode variations in pronunciation. Other annotations denote morphological properties. For instance, a "+" translates to a stressed root. For an example, the morph *car+* is a stressed root that occurs in the word *car.* But the morph *cAr+* is a stressed root that occurs in the word *caribou.* The determination of the morph inventory and the diacritics used have been fine-tuned during the course of our experiments, in an attempt to improve probability modeling and constraint[4]. In effect, an actual syllable is mapped to multiple morphs for different instantiations in varying word positions and spelling. The resultant expanded lexicon incorporates more linguistic context than that offered by the syllable unit alone, leading to enhanced constraint. In addition, here is a first step in merging pronunciation and spelling into the same representation. It will be clear later that this aids us in combining models for pronunciation and spelling so that ultimately hypotheses for letter-spellings can be accessed directly.

2.4.2 The Phonetic Network

The second feature of our system is the use of a phonetic network. More traditionally, word hypotheses from a first-stage recognizer are output into a word graph for processing by higher order language models in a later stage. The second pass is restricted to the words

[4] See Appendix B for a guideline on the morph notation.

hypothesized by the initial stage. Effectively, much of the search space has been eliminated irreversibly by the first-stage models. In our general vision, the output of the first stage is a subword network in which we are free to select the type of symbols situated on the graph. Indeed, the system may gain from reducing the morph hypotheses of the first stage back into arguably the most elementary atomic units of representation[5], the phonetic units. This is undertaken regardless of the set of morphs defined in the lexicon and proposed in the first stage. The network will encompass acoustic model and language model scores, the latter being derived from the morph lexicon. The network also omits all information pertinent to time. We advocate for the use of a phonetic network for several reasons:

- By returning to the phonetic unit, the second stage is not confined to the hypotheses stipulated in the first stage. This lessens the chance of committing irrecoverable errors by the first stage where the search space is significantly reduced.
- Morph constraints of the first stage exert influence via the scores embedded in the network. Although this perturbs the second stage search, the second stage is endowed with the flexibility to select any hypotheses favored by its own language models which can be entirely independent of the first stage.
- In building from phonetic units in the second stage, word-level language models can be applied in concert with sublexical models rather than using the word-level models alone. This is a means for coupling different models from the linguistic hierarchy within a single search but over a smaller search space. The second-stage search is described in Chapter 4.
- This scheme is more amenable to handling OOV words. When an unknown word occurs, the first stage should be equipped to produce the correct phones of the unknown word. This requires its models and lexical units to support the previously unseen phonetic sequence in question. But at this point we suspect that, with only partial knowledge, the actual morph sequence of an unknown should not be committed.

[5] Whether phonetic units are the most basic linguistic units in speech is a subject of contention in the literature. We have used phonetic units in the terminal layer of the ANGIE parse tree as these coincide with the basic units used in our speech recognizer acoustic models.

2.4.3 Lexical Generation

An implication of using the phonetic network output in stage one is that its lexical representation no longer needs to correspond with the domain-specific word lexicon of the second stage. Loosely speaking, the set of morphs in the first stage are not required to define or express the words of the second stage. Then, by breaking ties to the word lexicon, a key corollary is that the stage one lexical units can be redefined. What is most intriguing is that these units can be re-invented with a number of criteria in mind. These can pertain to optimizing constraint in the language models or promoting compactness. Algorithms can be applied to improve modeling in the first stage by synthesizing and fine-tuning the lexical representation.

More specifically, our philosophy is a generative one. We rely on learning for the system to generate its own lexicon, and it alone discovers automatically a more optimal but novel representation given the probability framework. A new lexicon is constructed given phonetic sequences of training data. One way to determine the added utility of a new lexicon is to consider the constraint of the models derived from it. This can be done by measuring perplexity, for instance. Given that it remains undetermined which set of units best capture syllable-level constraints, we believe that an automatically determined set of units may facilitate our modeling more so than one that is exclusively engineered by a hand-selection procedure. In Chapter 7, we detail how we begin with the original symbolic representations of the domain and depart significantly from them in an iterative algorithm. At each turn, perplexity is reduced. The algorithm is continued until convergence, indicating that the lexicon is "optimal" in the sense of perplexity reduction. An added benefit is also greater compactness in the overall representation. Chapter 7 will provide results of the generation algorithm and analyze the implications of those experiments.

2.5 A Comment on Stage Two

Our efforts have mainly focused on designing an effectively constraining low-level first stage. In the second-stage decoder, a combination of higher-level domain-specific linguistic knowledge is applied over a reduced search space. Historically, this decoder has evolved from earlier work by Lau [59] who made early attempts on flexible vocabulary speech understanding, using the ANGIE framework. In later chapters, we will elaborate on the second-stage architecture and experiments using this.

It should be noted that, in order to ultimately support a flexible vocabulary, this second stage needs to satisfy a number of criteria. In the following, let us briefly outline the ingredients of our second-stage design.

- Processing a phonetic network, the second stage primarily applies knowledge above the phone-level, up to and including the word-level.
- Ideally, it performs this by tightly integrating the information, derived from all the linguistic levels in use.
- As the decoder traverses a phonetic network, it inherently adopts a *bottom-up* philosophy in the search algorithm. That is, the recognizer refrains from applying word-level linguistic models and proposing any in-vocabulary word until the end of a phonetic sequence which may constitute a word. In this way, the acoustics and the low-level or phonotactic linguistic models steer the search so that OOV sequences are not prematurely pruned. A more conventional *top-down* strategy would allow higher order or n-gram models to favor the likely domain-specific words at the outset, pushing the unknown word hypotheses to the bottom of the stack where they risk being pruned. But in precluding early pruning, the bottom-up approach intrinsically invites a more expensive search; its success hinges on a more compact space to begin with. Our answer to this is a small phonetic network at the search input while the decoder undertakes the delicate balance between narrowing the search space as early as possible, and accounting for the possibility of OOV.

Chapter 3

The Representation of Hierarchical Sublexical Knowledge

3.1 Overview

This chapter will be a discussion on the relevant issues in modeling hierarchical sublexical knowledge. It will begin by introducing ANGIE and the underlying motivations behind its design philosophy. We will be visiting some theoretical claims that inspired ANGIE's model. This is followed by a detailed explication of the framework, including the grammar and the probability model. We will uncover the engineering obstacles which have prevented the full deployment of ANGIE in a speech understanding system. Next, we trace the development of finite-state transducers in computational linguistics, and discuss our vision of merging this technology with ANGIE to form the ANGIE-based FST. We debate the possible benefits that FSTs might deliver, and the routes one could take for transforming a hierarchical paradigm into the flattened FST. In the process, some of the ideas that have been conceived will be summarized, setting the stage for the experiments presented in later chapters. To conclude our discussion, we present one more beneficial outcome of our ANGIE-based FST: the integration of grapheme information within the recognizer.

3.2 An Introduction to ANGIE

First introduced in [97], ANGIE is a hierarchical paradigm for modeling sublexical phenomena such as phonology, syllabification and morphology, useful for speech and language applications. It is much inspired by theories that advocate the hierarchical organization of speech understanding, especially below that of the word level. The vision is to capture these phenomena in a data-driven computational engineering framework for speech applications. The framework would model language structures from the bottom up, sharing common information about internal structures of words, and encoding linguistic knowledge within and between each of the hierarchical layers. The result is trainable and probabilistic, modeling phonological processes by simple context-free rules. Its power and novelty lies with an ability to predict phone sequences of the language without explicit ties to a particular vocabulary, because the models are designed specifically to codify generic linguistic knowledge from data. The probabilistic models allow it to discover the linguistic patterns from training data and in-vocabulary sequences, and extrapolate that knowledge to previously unseen data. ANGIE was first applied to a letter-to-sound/sound-to-letter conversion system [69]. Since then, it has been used in the syllabification of large lexicons [79], hierarchical duration modeling for speech recognition [10] and word recognition experiments [59]. Encouraging results to date have motivated the work here, which extends past experiments further in several respects. Our work hopes to apply ANGIE in a near-real-time conversational interface for a real-world task, and aims to capitalize on ANGIE's power to model subwords, for unknown word scenarios, in an unprecedented fashion.

Before explicating the ANGIE paradigm and its applications to date, we would like to delve into the background issues that ANGIE has come to address. In particular, we highlight two issues that have been the driving factors towards the inception of ANGIE:

1. The modeling of sublexical phenomena such as pronunciation variation for speech applications, and

2. The use of a formal linguistic hierarchy in a computational approach.

3.3 Motivations of ANGIE

3.3.1 Sublexical Modeling

Sublexical modeling refers to the modeling of pronunciation variability of words. This is one issue that ANGIE strives to achieve. It also, as some researchers believe [66], forms a critical part of a speech recognizer, as pronunciation variability exists inherently among speakers and within speakers due in part to contextual effects. Moreover, words can have alternate pronunciations e.g., "either" may be pronounced as /iy dh ax r/ or as /ay dh ax r/[1]. Phonological effects are particularly difficult as they can be influenced by higher-level sublexical context such as syllable position or adjacent phonetic context. Thus, a sublexical model is required to predict these phenomena.

In the past, pronunciation graphs have been a popular means of sublexical modeling [77, 64, 109, 23, 14]. Based on techniques developed by phoneticians, these rely on rewrite rules that transform phoneme sequences into phonetic sequences in order to account explicitly for phonological effects, such as devoicing or flapping. Typically, they are pre-compiled into a lexicon and later expanded to yield a set of alternate pronunciations in the form of a graph. However, this method suffers some disadvantages. Generally, a large number of ordered rules is required to adequately capture allophonic variations within and across words. As the ordering is critical for producing the correct output, incrementally adding new rules is therefore difficult due to interactions that arise with existing rules. Researchers [103] have also attempted to attach probabilities to rule productions, that is, computing a probability estimate based on the number of times a rule is applied. But the probabilities generally assume that the ordered rules are independent of one another, which is clearly invalid. Sometimes, decision trees [88] are used to systematically generate phone realizations in context. Graphs consist of arcs connected via nodes, representing permissible phone sequences. Weights on the arcs can be computed via an iterative training process. A shortcoming of this method is the lack of sharing of common sublexical structures within the probability space, and this directly leads to a lack of robustness due to sparse data problems. It also demands retraining at every instant that a new word is added to the vocabulary.

Recognition systems which are based on HMMs [15, 62] do not confront pronunciation variability issues directly. They capture phonological processes implicitly by a variety of methods. Some examples are the use of context-dependent acoustic models such as triphones and the use of larger phonetic inventories that include units embedded with specific contexts.

[1] Throughout this thesis, we will use a modified ARPAbet nomenclature for phonetic units. These units will be depicted with enclosign "//"s.

3.3.2 Theoretical Background

The classical model of generative phonology [8] recognized only the segment as a structural unit in phonological representations. The syllable as a contributor to the organization of speech was notably absent. But compelling evidence suggests that phonological processes as well as suprasegmental phenomena such as stress and tone are described more succinctly with reference to syllable structure. In phonology, the last two decades have seen many proponents for the syllable as a hierarchical unit in phonological representation. These have included Clements and Keyser [13], Kiparsky [55], McCarthy [67], Selkirk [93], Kahn [51] and Fujimura and Lovins [21].

A formal definition of the syllable and rules for determining syllable boundaries have been sources of controversy. Selkirk [93] defines the syllable in terms of "well-formedness" on a sonority scale that ranks speech sounds according to their sonority, a notion related to glottal excitation and vocal tract opening [19]. For example, vowels are the most sonorant and voiceless stops are the least. It was hypothesized that each syllable contains a sonority peak, and a set of rules stipulates the syllabification. For instance, the Maximum Onset Principle attempts to maximize the number of consonants in the initial consonant cluster while Stress Re-syllabification assigns segments to the preceding syllable if it is stressed. An important notion is that the syllable is a hierarchical unit that can be internally decomposed into an *onset* (the initial consonant cluster) and *rhyme* (the rest). The rhyme is further subdivided into the nucleus and the coda (final consonant cluster). It was postulated that phonotactic constraints are tightest within these constituents. In his seminal thesis, Kahn [51] showed that a number of phonological processes such as flap formation, glottalization and r-deletion, which interact in intricate ways, can be resolved by a small number of simple statements when accounting explicitly for the syllable. He formulates rules for assigning syllable structure but stops short of proposing a general theory of phonotactics.

Church [12] carried on Kahn's theoretical proposition to construct a sublexical computational model based on a phrase-structure parsing technique. Phonological constraints are then expressed in terms of context-free phrase-structure rules. Church succeeded in discovering syllables for words by using phonological constraints, when given a phone sequence. This type of bottom-up computation was an early inspiration towards the conception of ANGIE because it validated that allophonic and phonetic cues are sources of information that can be utilized to recover syllable structure. Church also pioneered the use of context-free grammars to characterize subword phenomena, which is adopted by ANGIE. This plays an important role in easing the computational demand and therefore makes the application for recognition purposes a possibility. In Church's thesis, there is cursory mention that the same framework might be useful in the lexical retrieval component of a speech recognizer, and that augmenting with probabilities would cope with the inherent ambiguities of errorful recognition hypotheses.

In a similar vein, Meng's work [69, 70] used a hierarchy for modeling subwords but incorporated a probability model that could be trained. Letter-to-sound rules are encoded in a linguistic hierarchy that represented morphological, syllabic, phonotactic and graphemic constraints which act in concert. However, the framework was designed solely for the purpose of reversible letter-to-sound/sound-to-letter generation, with relatively error-free inputs. But not unlike our own visions, Meng pictured the concept of entering words orally into a speech recognizer and dynamically updating spellings and/or pronunciations, using the bi-directional functionality of her system. This initial work played a significant role as a predecessor to ANGIE which, by contrast, was designed with errorful recognition hypotheses in mind, and with the hopes of being incorporated into a real-time recognizer.

3.4 The Framework of ANGIE

3.4.1 Introduction

ANGIE [97, 59, 96] characterizes morphological and phonological substructures of words using a hierarchical representation, composed of multiple regular layers. This multi-layered representation is viewed as a parse tree, and is derived from a set of context-free rules that are designed by hand[2]. Overlayed on the hierarchical representation is a trainable context-dependent probability model. The dependencies are based on the neighboring and surrounding nodes on the ANGIE parse tree. ANGIE is designed to learn and characterize linguistic patterns pertaining to the internal structures of words of a language. In our case, we have only dealt with English thus far. Its underlying philosophy is to marry statistical modeling with a framework founded upon linguistic theories regarding the hierarchical arrangement of structures from the phone level up to that of the word. It is hoped that this

[2] The concept of learning and generating these rules automatically is beyond the scope of this thesis.

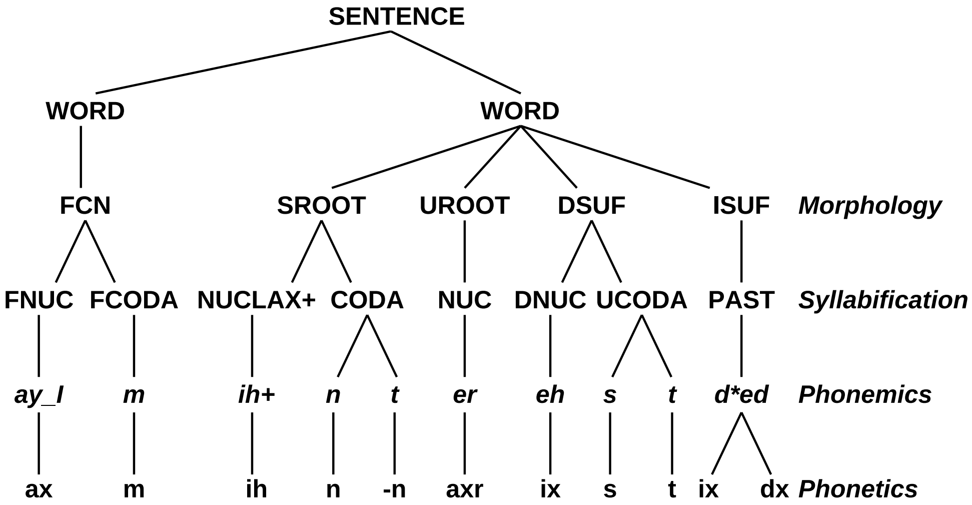

Figure 3-1: *Sample Parse Tree for the Phrase "I'm interested..." Below the word level are layers representing morphology, syllabification, phonemics and phonetics. At morphology layer are nodes for: function word (*FCN*), stressed root (*SROOT*), unstressed root (*UROOT*), derivational suffix (*DSUF*) and inflectional suffix (*ISUF*). At the syllabification layer are nodes for: nucleus and coda in a function word (*FNUC *and* FCODA*); vowel nuclei that are (1) stressed and lax (*NUCLAX+*), (2) unstressed (*NUC*), and (3) in a derivational suffix (*DNUC*); stressed and unstressed coda (*CODA *and* UCODA*); a past tense* PAST*. The phonemic layer contains pseudo-phonemic units with contextual markers such as "+" for lexical stress and word context such as "_I" and suffix context such as "*ed" for the past tense suffix. The terminal layer consists of phonetic symbols. A /-n/ marks the deletion of the parent phoneme /t/, tied to the left context /n/.*

knowledge can be exploited as linguistic constraints for speech recognition, especially for unknown words. The reason is that an ANGIE parse tree can be formulated from a sequence of phones from an unknown word and the probability model can produce a likelihood. Given an input sequence, the context-free grammar dictates whether a parse is permissible. If this is the case, a hierarchical structure can be obtained and the probability model generates a corresponding score, assigning low scores to parses which seem less likely to occur in English, in accordance with the model.

In the next sections, we expound on the elements that make up the ANGIE framework. In assemblying a system using ANGIE, it is necessary to perform the following:

- **Define the grammar.**

  A developer needs to define a set of context-free rules covering the patterns of subword structures. This will embed the linguistic knowledge that we attempt to utilize. We discuss the grammar in Section 3.4.2.

- **Define the lexicon.**

  ANGIE requires a lexicon which defines the set of in-vocabulary words for training and learning the subword patterns. The lexicon is arranged in a two-tier structure, discussed in Section 3.4.3.

- **Train the ANGIE models.**

  We briefly overview the parsing operation and the probability models in Section 3.4.4. More information regarding the training and search procedures, and further engineering issues were investigated in [59]. We choose not to detail these issues here as they have not been the focus of our work.

Detailed examples will be drawn from the grammar implemented in the JUPITER domain. The development of this grammar has constantly evolved through iterations over training data from JUPITER but the characteristics of the grammar itself remain domain independent.

3.4.2 The Grammar

Depicted in Figure 3-1, is an ANGIE parse tree example representing the partial utterance, *"I'm interested ..."* The context-free grammar, which mandates the tree structure, is written such that, when a left-hand category corresponds with one layer, the right-hand categories refer to the layer immediately below. This forms a very regular multi-layered structure. In total, there consists of six layers. The top layers contain the WORD and SENTENCE nodes[3]. The remaining layers from the bottom up refer to phonetics, phonemics, syllabification and morphology. The bottom terminal layer can dually represent letters or phones, depending on the application[4].

The phonetics layer contains the most basic segmental units for recognition, the phone set. In the past, these have directly corresponded with the acoustic models applied to the speech signal during recognition. However, as the acoustic models are ultimately independent of the ANGIE language model, our recognizer now utilizes context-dependent diphones. The phonetic segment hypotheses are extracted from these. In addition, a "-" preceding a phone, as in the /-n/ in *"interested"* shown in Figure 3-1, functions as a placeholder signifying the incidence of deletion. In the case presented here, the */t/*[5] phoneme following the */n/* has been deleted[6]. There are approximately 100 units at the pre-terminal phoneme layer. They can be considered "pseudo-phonemes" because these phonemic units have been annotated with additional linguistic context. These phonemic units embody distinctions among the following characteristics:

- Stressed vowels marked by a *"+"* versus unstressed vowels, e.g.*/ih+/* is the phoneme */ih/* in a stressed syllable.
- Consonants in syllable onset position denoted by *"!"* versus those in non-onset position, e.g.*/b!/* for a */b/* in onset position.
- Units specific to certain inflectional suffixes, e.g. */d*ed/* for */d/* in past tense.
- Units specific to certain function words, e.g. */ay_I/* for */ay/* in the word *I*, as seen in Figure 3-1.

Furthermore, there are additional "double phonemes" such as */nk, nt, ts/* and "pseudo-diphthongs" such as */er/* shown in Figure 3-1. More examples are */ihl, aer/*.

[3] These two categories are simply place-holders, but could later be replaced by alternatives such as syntactic/semantic units which may also have a bearing on pronunciation and stress patterns of words.

[4] Past implementations have allotted the terminal layer for grapheme units for the purposes of letter-to-sound/sound-to-letter conversion. By contrast, this thesis will only employ phonetic units as terminals, as these are the basic acoustic units used by our speech recognizer. But we explore other means for incorporating grapheme information.

[5] Phoneme units from the preterminal layer are enclosed in "//" in the text. To distinguish from phonetic units of the terminal layer, phoneme units are italicized. Refer to Appendix B for a comprehensive explanation of all notation used in this thesis.

[6] We consider the deletion to have occurred when acoustic evidence for the */t/* phoneme is absent or too weak to have been detected by the recognizer.

Processes that govern phonological rules are captured by the phoneme-to-phone layer. From Figure 3-1, the following context-free rule has instantiated:

$$er \;\; \Rightarrow \;\; \mathrm{axr}$$

This states that the unstressed diphthong /*er*/ has reduced to a retroflexed schwa. Under this scheme, the probability model is set to learn context-specific pronunciation rules from the training data. For instance, consonants in the syllable onset position are subject to different constraints from those in the coda position. Phonological reduction is also more likely in function words, which are therefore modeled separately.

During development, the phone and phoneme sets were determined and have evolved based on a number of engineering decisions, made in the interest of overcoming sparse data problems together with optimizing the probability models. In fact, the actual units chosen were fine-tuned via many iterations of closely examining recognition outputs and forced alignments.

The syllabification layer describes sub-syllabic structures; examples include ONSET, NUC and CODA. Special units are defined for structures in various contexts. These include function words, for example FNUC and FCODA, as in Figure 3-1. In some cases, distinctions are made for stressed and unstressed syllable, such as CODA versus UCODA. In other cases, the context is dependent on the above morphological layer, for example DNUC for a vowel nucleus under a derivational suffix, and PAST for an instantiation of an inflectional suffix. Additionally, we can place context dependencies stemming from the layer below: an example is NUCLAX+ for the subset of stressed vowels that are lax[7].

The morphology layer governs the word's morphemic breakdown. All words either contain a stressed root (SROOT) or are categorized as function words (FCN). For non-function words, morphological units include prefixes (PRE), unstressed roots (UROOT), "derivational" suffixes (DSUF) and "inflectional" suffixes (ISUF)[8].

As lexical stress potentially influences multiple levels of the linguistic hierarchy, its information is distributed throughout the morphology, syllabification and phonemics layers

[7] See glossary in Appendix A for a definition.

[8] To a first approximation, ISUF is a category which, if omitted, would leave behind a fully formed lexical entry. Our utilization of DSUF is pragmatic, and only loosely affiliated with the formal linguistic definition.

Word Lexicon		
barbados	:	bar+ bAd+ -os
islamabad	:	is- lam+ a bAD+
predict	:	pre- dict+
predicted	:	pre- dict+ =ed
cambridge	:	cAm+ -brige
camden	:	cam+ -den

Morph Lexicon		
Stressed Roots		
baD+	:	_b! aa+ d_
bAd+	:	_b! ey+ d_
bar+	:	_b! aar+_
cAm+	:	_k! ey+ m_
cam+	:	_k! ae+ m_
dict+	:	_d! ih+ k t_
lam+	:	_l! aa+ m_
Prefixes		
is-	:	_ih z_
pre-	:	_p! r iy_
Derivational Suffixes		
-bridge	:	_b! r ih jh_
-den	:	_d! en_
-os	:	_ow s_
Inflexional Suffixes		
=ed	:	_d*ed_
Unstressed Roots		
a	:	_ah_

Table 3-1: *Example of a Two-Tier Lexicon of ANGIE. See text as well as Appendix B for explanation of the meanings of the diacritics.*

by explicit "+" markings. Currently, lexical stress is not included in the terminal layer. Our preference is to omit explicit stress markers from the acoustic model units, and to allow the models to discover tendencies for phonetic realizations related with stress patterns. One example is schwa reduction.

3.4.3 The Lexicon

The ANGIE lexicon is arranged in two tiers: first the pseudo-phonemic sequences define a set of morphemic units and secondly, words are given by their morphemic baseforms. Table 3-1 gives some examples of baseforms taken from the two-tier lexicon. Multiple alternative pronunciations are also allowed in each of these baseform definitions[9]. During the recognition process, lexical access takes place at the phonemic layer. It was alluded to earlier in Section 2.4.1 that a possible set of units for a recognizer lexicon are syllable-based, and additionally codify spelling and pronunciation. We have utilized precisely these in the intermediate morphemic lexicon[10]. The following summarizes their characteristics:

- Morph units encode positional constraints with a set of diacritics. These tie them directly with nodes from the morphological layer in the ANGIE tree. For example, the morph appended with a "-" denotes a prefix such as *re-*. Meanwhile, context-free rules serve to restrict the prefix to only occur in the initial syllable position.
- Letters of the morph correspond with the actual spellings.
- Capital letters are a device to distinguish between homomorphs.

Although ambisyllabicity poses a major unsolved issue whenever one is using syllable-level lexicons, our working philosophy is to choose the morph or syllable boundaries in accordance with the dual principles of (1) striving for consistency and (2) ensuring minimal sparse data problems. Note that, in many instances, the actual syllabification may differ from more popular choices advocated in linguistic theory, but our decisions are justified by the abovementioned engineering principles. Compared with using a single lexicon of phonemic sequences for each word, a two-tiered lexicon can help distinguish ambiguous parses and supply improved constraint. This is because the morphs dictate the placement of the boundary positions as well as other linguistic context.

3.4.4 The Dynamic Parse Mechanism and Probability Models

The parsing operation proceeds in a bottom-up and left-to-right manner, in a breadth-first search. Given an input sequence of phone terminals, it tries to generate one or more parse trees. The details are given in [59].

During the same process, the parser also applies the probability model. Traditionally, context-free rule formalisms incorporate probabilities based on rule production frequencies. But ANGIE's probability distributions are *spacio-temporal*, and are designed intentionally to

[9] Multiple pronunciations can be modeled either as phonological processes or offered as alternatives at the baseform definition level. The choice is made by the grammar developer based on engineering decisions to optimize the predictive power of the models.

[10] It will be seen later that the same morphs form the first-stage lexicon, and are therefore conveniently well-matched to the ANGIE sublexical model which operates above the n-gram model.

<table>
<tr><td colspan="12">SENTENCE</td></tr>
<tr><td colspan="2">WORD</td><td colspan="10">WORD</td></tr>
<tr><td colspan="2">FCN</td><td colspan="3">SROOT</td><td>UROOT</td><td colspan="4">DSUF</td><td colspan="2">ISUF</td></tr>
<tr><td>FNUC</td><td>FCODA</td><td>NUCLAX+</td><td colspan="2">CODA</td><td>NUC</td><td>DNUC</td><td colspan="3">UCODA</td><td colspan="2">PAST</td></tr>
<tr><td>*ay_l*</td><td>*m*</td><td>*ih+*</td><td>*n*</td><td>*t*</td><td>*er*</td><td>*eh*</td><td>*s*</td><td colspan="2">*t*</td><td colspan="2">*d*ed*</td></tr>
<tr><td>ax</td><td>m</td><td>ih</td><td>n</td><td>-n</td><td>axr</td><td>ix</td><td>s</td><td>scl</td><td>t</td><td>ix</td><td>dx</td></tr>
</table>

Figure 3-2: *Tabular Schematic of a Parse Tree. The phrase is I'm interested ..*

be able to predict the probability of the next input unit given the very localized history. They are determined by the internal node relationships, and are made possible by the regularity of the parse tree. The parse tree can be treated as a table which consists of a sequence of vertical columns. The tabular format depicted in Figure 3-2 is equivalent to the parse tree from Figure 3-1. A *column* can be defined as the nodes along the path from the root to a leaf node of a tree while each row of the table represents a layer of the parse tree. Traversing the tree from left to right involves sweeping the table from one vertical column to the next. Two types of probabilities are compiled:

1. *Advancement Probabilities:* These are the conditional probabilities of a leaf/terminal phone node in the parse tree given its entire immediate left column context.

2. *Trigram Bottom-Up Probabilities:* These conditional probabilities are specified for internal nodes which are non-terminal units in any layer above the bottom most. Within the parse tree, each internal node's probability is given by the context of its immediate left sibling node and its immediate child node. For example, from Figure 3-2, $P(\textsc{fcoda}|\textsc{fnuc}, /m/)$ specifies the probability of the FCODA[11] node.

By summing the log advancement probability and all log trigram bottom-up probabilities for the current column (up to the point where it merges with the left column), we yield a *total column score*. The probability score of a parse table or tree, which essentially represents the score for a single word or an entire sentence, is computed as the sum of the individual column scores. For reasons of sparse data, the advancement probability between columns at a word boundary is simply computed as a phone bigram estimate conditioned on the

[11] This node can be interpreted as the part of the syllable in coda position and one that only occurs in function words.

word transition context. At other points of sparse data, standard smoothing techniques are incorporated.

The ANGIE column can be viewed as a phone unit embedded with long-distance contextual information encoded in the upper layers. From such a viewpoint, the advancement probability is highly constraining. Essentially, the probability models will learn phonological rules from the phonetic realizations, given a host of contextual properties, guided by the context-free rules. For instance, in Figure 3-2, the trained grammar will learn that the */t/* phoneme is likely to be deleted following a */n/* phoneme in the coda position. This information is derived from:

$$P(/\text{-n}/|\textsc{word}, \textsc{sroot}, \textsc{coda}, /n/, /\text{n}/)$$

and

$$P(/t/|/n/, /\text{-n}/)$$

The grammar also learns the subsequent reduction of the retroflexed vowel by:

$$P(/\text{axr}/|\textsc{word}, \textsc{sroot}, \textsc{coda}, /t/, /\text{-n}/)$$

These models together encode implicitly the rule that */t/* can be delted in the coda position after */n/* and before a reduced vowel. The training procedure involves first creating phonetic-orthographic sentence pairs for the corpus. Usually, this is done by computing forced alignments seeded from an initial speech recognizer. Lau [59] measured the per phone perplexity of ANGIE on ATIS flight domain data, and found that ANGIE performs better than a trigram on unseen test data.

3.4.5 Discussion

A major novelty of the ANGIE framework is its ability to generalize towards new or previously unseen linguistic patterns from the trained grammar. Sublexical phenomena are initially specified by the hand-written rules for the grammar, and their manifestation is documented during training. While any parse tree not licensed by the grammar will not be permissible, the grammar itself encompasses a large number of parse trees. The probability model is relied upon to favor the correct parses, by assigning higher scores to them, and also

WORD			WORD			
STRESSED ROOT		ISUFF	STRESSED ROOT			
ONSET	NUC+	PLURAL	ONSET		NUC+	CODA
_d!_	_ey+_	_s*pl_	_p!_	_l_	_ey+_	_s_
d	ey	z	p	l	ey	s

Figure 3-3: *Tabular Schematic of ANGIE Parse Trees. The words represented are days and place.*

automatically learn the more subtle constraints that are peculiar to the context-sensitive properties of English phonology, not explicitly covered by the context-free rules. Most importantly, the framework is intended to enable knowledge gathered at the training phase to generalize towards previously unseen sequences upon testing. Generalizing particularly benefits speech recognition tasks because the probabilistic framework will tolerate errorful input phone hypotheses without parse failure. Yet errors will be discouraged by low probabilities, as is required by a recognition algorithm. This *over*-generalization actually relies on the ability to share common word substructures by way of common subtrees in a parse. Probabilities are pooled across training data for similar substructures within different words or, alternatively speaking, sublexical phenomena with similar contexts.

Consider the following example. The words *preventable* and *predictable* share a common prefix *pre-*, and thus will share frequency counts that correspond with their respective common subtree structures. The model will learn that *pre-* is a frequently occurring prefix; it will learn the phonological rules that tend to occur within this subword; and it will also learn to some extent the types of structures which are likely to follow this prefix[12]. Consequently, rare words can benefit from observations of common words that have the same local phonetic environments. Phonological rules learned from more common words can be applied to rare words with full probability support.

Moreover, words that are completely unknown to the recognizer can be generated with a non-zero probability as long as the parse is admitted by the grammar rules, and the subtree fragments, representing localized sublexical patterns, are individually supported with non-zero probability. More specifically, let us examine two more examples depicted in Figure 3-3.

[12] The probability framework restricts the model to predict only the following column from the previous left. But much redundant information associated with linguistic context resides within these columns, hence revealing even longer distance linguistic information.

Shown are the parse trees (in tabular format) for the words *days* and *place*. If these two parse trees are instantiated in the training corpus, other words with tree structures partially in common with these words can share training data. For instance, the parse tree for the word *plays* is illustrated below in Figure 3-4. ANGIE's probability models can predict this parse entirely from the training instances of *days* and *place*. This is because all the model components, that is, the advancement and trigram probabilities, can be extracted from those within *days* and *place*. This can be achieved even with zero instances of the word *plays*.

WORD			
STRESSED ROOT			ISUFF
ONSET		NUC+	PLURAL
p!	*l*	*ey+*	s*pl
p	l	ey	z

Figure 3-4: *Tabular Schematic of ANGIE Parse Trees. The word depicted is plays.*

In principle, ANGIE can (1) provide a linguistic score for OOV words that are encountered during recognition and (2) easily acquire new words without lexical retraining, as probabilities for phonological rules can be leveraged from the trained models for existing words. These capabilities have been envisioned since the beginning of ANGIE's inception. And our research aims to exploit these features in a flexible vocabulary system.

One critical caveat is that ANGIE's power to *over*-generalize from a finite amount of training data translates to a coverage spanning an infinitely large probability space. This is attributable to the probability framework. For a single word, there are multiple ambiguous parses, and the total number of parsable words covers an arena much wider than even that of general English. It is up to the probability models to exercise preference for parses most likely to conform to an English vocabulary word. Thus the parsing operation is required to traverse a large space in order to find the correct parse. In past experiments [59], even when exclusively recognizing in-vocabulary sentences, this has proven to be too computationally demanding for integration with a real-time recognizer. We will discuss this obstacle further and propose an alternative representation in Section 3.7.

3.5 Past Applications of ANGIE

Before furthering our discussion on new ideas for applying ANGIE, we devote this section to reviewing previous results. Past successes have highlighted ANGIE's strength as an effective multi-purpose tool that encapsulates sublexical phenomena.

3.5.1 Letter-to-Sound

The first application of ANGIE was in letter-to-sound generation [97], quite similar to the experiments conducted with its predecessor in [70]. The probabilistic parsing algorithm was used to parse letters of an input word in a breadth-first search, and the pronunciation was derived from the phoneme sequence at the pre-terminal layer. Experiments were conducted on the Brown corpus and a test set accuracy of 68.9% per word and 89.7% per phoneme was achieved. The system was also used in sound-to-letter generation. Here 53.2% per word and 88.5% per phoneme accuracy on a test set was achieved.

3.5.2 Duration Modeling

ANGIE has also been applied to a probabilistic duration model designed to enhance speech recognition [11, 10]. The hierarchical framework captured duration phenomena at multiple levels of the linguistic hierarchy simultaneously. At the core of the idea was a normalization scheme, performed on the ANGIE parse tree nodes, which accounted for durational variability at each successive level of the tree. This strategy was very effective at dealing with sparse data problems and yielded both a robust measure of rate of speech and duration models that were normalized with respect to speaking rate. In addition, this framework was used as a basis for exploring and discovering speech timing phenomena such as the secondary effects on relative duration due to variations of speaking rate, the characteristics of anomalously slow words and prepausal lengthening effects. In phonetic recognition, a relative improvement of up to 7% (from 29.7% to 27.4% phone error rate) was achieved. In word-spotting, the addition of a duration model increased performance from 89.3 to 91.6[13]. These encouraging results demonstrated the utility of durational information for recognition applications. As a consequence, it was concluded that the hierarchical paradigm was very compatible for

[13] These numbers were quoted using a standard metric for word-spotting called *Figure of Merit.* See [10].

capturing the nature of segmental effects on duration, and, in fact, the application could be extended to other types of prosodic phenomena.

3.5.3 Subword Modeling and Flexible Vocabulary

In his thesis [59], Lau applied the ANGIE framework in a number of recognition experiments trained on ATIS data [112]. In an effort to focus on assessing the sublexical model itself, Lau first implemented a phonetic speech recognizer which, in entirely omitting the word lexicon, circumvented immediate computational issues. Although a lexicon was not used, it implicitly trained the ANGIE subword models. The ANGIE parser proposed pseudo-words bottom-up from the rules, and periodically proposed completion of a word, at which point scores computed from higher-level language models were added. ANGIE achieved a phone error rate of 36% against a baseline result of 40%, where the baseline system utilized a phone bigram. These positive results were attributed to both the improved phonological modeling and the more powerful language model in the upper layers of the hierarchy.

Given that encouraging results were ascertained, Lau proceeded to explore the impact of varying sublexical constraints for word-spotting tasks where ANGIE provided the constraints for the keyword as well as the filler space [60]. The task was to spot city names in the ATIS domain. Results validated that greater sublexical constraints imposed by ANGIE in the filler model delivered better word-spotting performance than constraints supplied by a phone bigram model. Again, this would indicate that ANGIE is effective in modeling phonological rules probabilistically where the words are not known, such as in the filler of a word-spotting task.

The next application for ANGIE to tackle would naturally be continuous word recognition. This was accomplished in [61]. Although, results were comparable to baseline, a significant performance improvement was not found to be the case, and the computational load decreased speed significantly. More importantly, Lau was exploring the integration of the ANGIE framework with TINA [95], our context-free grammar based NL understanding system. Combining the bottom-up sublexical model of ANGIE with the top-down supra-lexical model of TINA into a single search, Lau implemented a stack decoder which consults the two parsers for scores and maintains total scores in a stack of partial paths. The NL component offers feedback at every putative word ending, instead of at the end of a sentence, as is conventionally the case. This ANGIE-TINA configuration yielded a 21.7% error rate reduction when compared with a system using a word bigram only. In comparison, if the NL constraints were used via the N-best resorting method, a smaller improvement (18.7% error rate reduction) was found. This work utilized phonetic networks, an extension of the conventional method of generating word graphs, that pruned down the search space after an initial "fast match" stage. This core method has been carried through to our work in two-stage systems which are also interfaced with phonetic networks. Unfortunately, Lau never solved fundamental computational issues, despite results that established the benefits of ANGIE sublexical models for both in-vocabulary and unknown words. He was grappling with the problem of devising an effective search strategy while controlling the computational complexity in terms of space and time, which was necessarily large due to the number of competing ANGIE theories generated. See [59] for details. To this end, his architecture, which preceded the availability of an FST-based system, seemed unwieldy and impractical for real-time recognition. Moreover, little success was obtained in exploiting syllable-sized units which were mentioned but not extensively investigated.

As an extension to his experiments in ATIS, Lau made first attempts to demonstrate the addition of new words dynamically to a recognizer vocabulary. A set of city names in the corpus were set aside as "unknown." In the baseline recognizer, using a pronunciation graph, lexical arc weights were set to zero for the "new" words. By comparison, in the ANGIE system, models were trained with the data, omitting the set of "new" words. It was hoped that ANGIE would support the lexical probabilities of the new words without actually training on them, facilitating the recognizer to propose them anyway. That is, given the transcription of a new vocabulary word, ANGIE should not require additional training data to support its pronunciation models. Unfortunately, the results did not surpass those of the baseline, nor could it be established that ANGIE's results were superior to those of a standard pronunciation graph. In any case, this experiment involved simulating highly artificial conditions because it failed to address the most challenging aspects of a flexible vocabulary system: detecting the presence of a previously unseen word, and automatically acquiring its baseform and spelling.

The above experiments have set the stage for our continued development of ANGIE applications in recognition. While prior experiments asserted the feasibility of the ANGIE framework and grappled with pruning and control strategy issues, we will direct our efforts towards assembling a workable multi-domain architecture for real-world applications. Using real data, we set out to explore optimal ways for integrating hierarchical constraints, which, for the first time, can be accomplished at manageable computational speeds. As will be seen, our more sophisticated methods of integrating ANGIE leave us better equipped to utilize its powerful models for a flexible vocabulary system.

3.6 Finite-State Transducers

Finite-state devices [1], such as finite-state automata, graphs and finite-state transducers, have been extensively used throughout the field of computer science. For many years, finite-state devices in computational linguistics were regarded as inferior to the more powerful context-free and unification grammars. But recently, we have witnessed a re-emergence of finite-state technology due to advances in the mathematical algorithms [90]. With a growing number of applications in natural language processing, finite-state automata are increasingly being adopted for speech recognition. They enhance the performance of search algorithms by offering uniformity and efficiency when used to combine multiple information sources with disparate representational units.

Finite-state machines (FSM) or finite-state automata (FSA) are automata that can accept regular languages only. An FSM contains a finite number of states and a function that determines transitions from one state to another as symbols are read from an input. The machine starts at an initial state and the input is positioned at the first symbol of an input string. The machine transitions from state to state as we proceed across inputs of a string until the end, and we finish at one of the designated final states. A *transducer* is an extension of an FSM in that it has the added feature of outputting a symbol upon transition from a state. At the termination of input, an output string is produced. This constitutes the mapping of symbols of an input alphabet to symbols of an output alphabet. Upon every transition taken, a *weighted* FST also emits a cost for taking the transition. In our recognition models, this represents a probability score. The mathematical definitions of FSAs/FSMs and FSTs are given in the glossary in Appendix A.

One primary feature is that FSTs admit a *composition* operation which allows the combination of multiple complex transducer mappings into one transducer structure and ultimately one operation. And furthermore there exists a collection of standard algorithms for manipulating and optimizing these transducers. For example, the *determinization* algorithm eliminates large redundancies that often prevail in FSTs used in applications such as speech recognition. Non-determinism refers to the phenomenon where multiple outgoing arcs with identical input labels emanate from a single state. As in most applications, a suitable path through a large transducer needs to be found efficiently, but non-determinism will adversely affect the speed with which the transducer can be searched. Following the application of the determinizing algorithm, every state in the transducer will have at most one transition labelled with a given input string, thus reducing the time and space required to process the input sequence. Mohri and Riley [71] reported large improvements in running time due to applying determinization. Pereira et al. [83] also relates a weighted *minimization* algorithm that collapses further redundancies in the transducer structure, reducing the storage size of an FST. The algorithm first redistributes weights towards the initial state as much as possible, followed by the application of classical automata minimization.

In speech recognition, FST-based recognizers [89] have been successfully implemented for medium-sized vocabulary tasks and proven to offer many advantages. They present a uniform representation for information sources and data structures such as context-dependent units, pronunciation dictionaries, language models and search lattices. Concisely, the recognition task is modeled as a composition:

$$T = (S \circ A) \circ U \tag{3.1}$$

$\circ$ is the operator representing FST composition of two FSTs. S represents the acoustic segmentation graph[14]. A represents the acoustic model scores based on observations at recognition time. During the search, the algorithm performs a composition between $(S \circ A)$ with U which is treated by the search as a single FST. In fact, U is a successive cascade of transductions from acoustic labels through to the language models. U is a complete model of the search space, and can be the transduction of any number of models and knowledge sources used in recognition. Usually, it is computed as follows:

$$U = C \circ P \circ L \circ G \tag{3.2}$$

where C maps context-dependent labels to context-independent labels, P applies phonolog-

[14] This is strictly applicable for segment-based recognizers. S is generally not included in the standard literature.

ical rules, L is the lexicon mapping pronunciations to words, and G is the language model. Any of these transductions can be weighted. At the final composed FST, an observation sequence is transduced to the sentence hypothesis with some negative log probability, computed as the sum of log probabilities associated with the various intermediate transducers. While, practically speaking, it is sometimes not possible to pre-compose the entire transducer U, the composition operations are performed *a priori* on some or all of the various language constraints, and additional knowledge sources may be composed "on-the-fly" at recognition run-time. For a pre-composed FST, subsequent optimization ensures that U can be searched as efficiently as possible, leading to a faster search. Alternatively, "on-the-fly" operation affords greater flexibility and requires less memory storage. The role of a decoder is to find the best path through the transducer U, which is treated as an ordinary search lattice.

In our experience, pre-loading a pre-composed and optimized FST affords excellent efficiency for run-time operation, but requires some craftsmanship, especially when attempting to compose together several cumbersome FSTs. In general, at each composition stage the following operations are undertaken:

$$min(det(A \circ B))$$

But the risk of failure at determinization persists because this algorithm expands the number of arcs, at each time when several arcs of the same input label exit a single node. Upon failure, of course, further composition cannot proceed. In addition, determinization itself directly impacts on the search speed. Its success can be critical in the practicality of a system. Hence, designing determinizable FSTs is an integral issue.

The FST framework is an intrinsically flexible and powerful framework in that multiple knowledge sources can be applied at different cascade levels and in novel ways. Intermediate models may have widely different sets of representational units. Yet from the point of view of the search algorithm, there is one uniform and optimized network.

In this thesis, we shall see that the FST, with its algorithmic advantages, offers a solution for approximating ANGIE models, by folding the rich hierarchical knowledge into a flattened data structure, thereby easing computation. It gives us the opportunity to experiment with different levels of sublexical constraints, and to incorporate novel sets of intermediate lexical units into our recognition engine.

3.7 ANGIE and the FST Representation

Section 3.4.5 first raised the issues of computation associated with employing a hierarchical model such as ANGIE in a recognition scheme. And past attempts [59] have persistently encountered computational issues. The problem stems from using the dynamic parser which inevitably involves a computationally intensive search process. This then is only feasible when conducted on a limited subset of the search space, thereby preventing the full exploitation of ANGIE's benefits. These issues are particularly pressing for us because our objective is to use ANGIE maximally to support both in-vocabulary and OOV words. The availability of a recognizer using the FST data structure opens up the possibility of translating ANGIE's models towards a flattened FST format[15]. Cast in this light, new questions will arise regarding how the transformation can best be accomplished.

FSTs have proven their versatility through the years, with extensive applications in computational linguistics, namely in dictionaries, morphology, phonology and so on. In the field of capturing probabilistic context-free grammar (CFG) formalisms in an FST, there has been much previous work. FSTs can represent regular languages exactly, and approximations to phrase structure context-free grammars are commonly used [90]. Unfortunately, because we have chosen to replicate the unique probability structure of ANGIE, we cannot simply draw upon these well-established algorithms. Nonetheless, the FST seems to be a natural mode of representation for us.

Consider the issues at hand for designing an FST that can express ANGIE's language constraints. We are faced with the need to account for two conflicting interests:

1. A compact or tractable representation of the vast probability space.

2. Optimal coverage of ANGIE's probability space.

The major obstacle is that, from a practical standpoint, it is impossible to output all allowable ANGIE parses within a single FST of reasonable size. It simply necessitates an algorithm that approximates ANGIE's coverage by selecting a subset of the space. In doing

[15] The FST recognizer which we have employed is outlined in Chapter 5. More information regarding the development of the core JUPITER recognizer adopting an FST paradigm can be found in [110].

so, we must consider how ANGIE's probability likelihoods are best emitted. On the other hand, restricting the coverage compromises the original unique features that are central to ANGIE. That is, the flexibility required to support previously unseen OOV words may be hampered. Hence, the key concern is how to optimally engineer the trade-off of outputting a portion of the probability space and emitting partial ANGIE parse information.

From the outset, our objective has been to devise a medium where ANGIE is folded into an FST, fulfilling all the above qualifications. Naturally, there exist many alternative solutions. Some of these have been given careful consideration in this thesis. But first of all, a solution will need to determine the following factors:

- General structure of the FST.
- Output symbol representations. This includes the problem of how the internal structure of a parse tree can be accessed upon conversion to the FST structure.
- Probability assignment on the FST arcs. Scores assigned to the FST arc weights exactly replicate ANGIE's probability estimates of each parse. As probability values perturb the paths taken during search, the distribution of these scores impacts upon the search efficiency and outcome.
- The algorithm for generating the FST. We have chosen to begin with a fully trained set of ANGIE models and explore different ways of transforming them into a single FST. All the algorithms described here will involve starting with ANGIE and subsequently generating an FST from the grammar. A central issue to the generation algorithm is coverage of ANGIE's probability space. We expound on this at length further on.

It is also assumed that the FST input alphabet will be the phonetic sequences proposed by the acoustic models. In the following, we provide a synopsis on the two main ANGIE-FSTs which were implemented in our experiments. The algorithms used to generate these are detailed in Chapters 5 and 6. The experiments evaluating their relative utility are also described there.

3.7.1 A Tree-like Graph Structure

In Chapter 5, our first attempt is to construct an FST as a tree-like structure. Being a tree that represents all the alternative pronunciations of vocabulary words, this resembles the commonly used pronunciation network. And it seemed to be the most obvious way

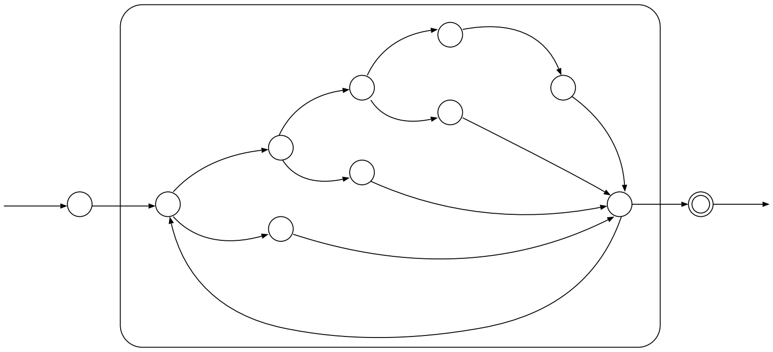

Figure 3-5: *Schematic of a Left-to-Right Branching FST. Words with common phonetic sequences advancing from the left, will share common arcs.*

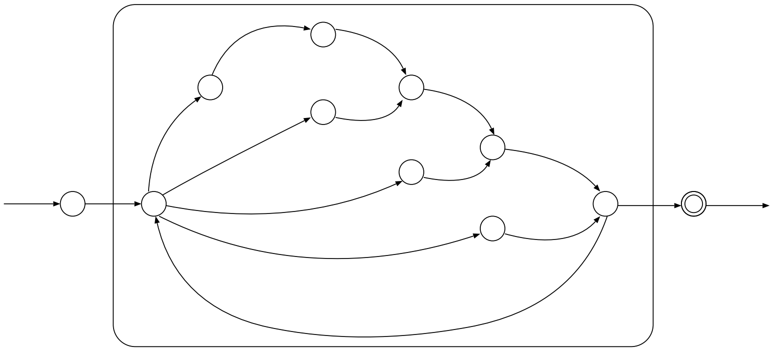

Figure 3-6: *Schematic of a Right-to-Left Branching FST. Outputs are emitted from the left, and arcs are successively merged together from left to the right.*

to capture ANGIE which in itself essentially models pronunciation. For such a structure, all phonetic sequences that were realized in the training data are observed and collapsed together. This may either be in a left-to-right branching style, illustrated in Figure 3-5, or in a right-to-left branching style, illustrated in Figure 3-6. In the left-to-right style, words with common phonetic sequences advancing from the left, will share common arcs. Reading from the left, branching occurs where the phonetic sequences first differ for the particular words. Outputs are emitted at the end of the sequences or at the right most arcs. In a right-to-left style, outputs are emitted at the beginning in time, and arcs are successively merged together from left to the right. That is, common phonetic sequences share arcs from the right hand side.

The philosophy of a tree representation rests upon building a collapsed data structure after examining the training data and assigning pre-computed ANGIE probabilities on the arc weights. The outputs are the vocabulary from a fixed lexicon. The compactness is gained from collapsing identical partial phone sequences together onto the same FST nodes and arcs. But these sequences often differ in ANGIE parse structure so that essentially, this strategy amounts to the loss of parse information. In fact, the entire process relies on memorizing training data instances, so that sequences that have not been realized in the training corpus are excluded from the FST. Despite having been allocated probability within ANGIE itself, they are not admitted by the FST.

In Chapter 5, we eventually select the right-to-left branching implementation because the outputs are emitted first. This results in a significantly more tractable computation during the search, as the n-gram language model scores are evaluated early. However, ultimately, the final composed FST overwhelmingly biases the recognizer towards exclusively recognizing fixed vocabulary items. In all, it turns out that this methodology fundamentally relied too much on the training data and did not reflect ANGIE's ability to generalize at all. A thorough discussion and an analysis of alternatives are offered in Section 5.7.

3.7.2 An FST Structure Based on ANGIE Columns

At this point, we must seek an alternative which does not bind sequences to those of a fixed lexicon. One could conceive of an FST which allows alternative pathways for OOV sequences in addition to in-vocabulary data which have been generated from training in a similar manner as that described above. This is illustrated in Figure 3-7. Ultimately,

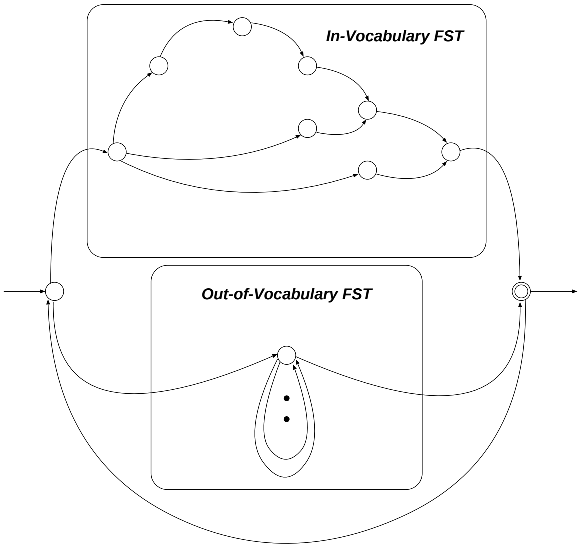

Figure 3-7: *Schematic of One Type of FST Supporting OOV Words. There are alternative pathways supporting OOV sequences in addition to in-vocabulary data.*

the recognizer opts to enter the search space where sequences belong to known words or a space for OOVs. The latter can be a phone loop, among other ways to model OOVs as opposed to in-vocabulary words. Ultimately, such a methodology does not embrace an ANGIE philosophy of modeling phonetic sequences from the bottom upwards; that is, treating OOVs and in-vocabulary words the same way and using general sublexical knowledge.

Cast under these considerations, we are driven to ponder on FSTs that capture the ANGIE parse structure succinctly. It seems then that some of the compactness must be sacrificed in order to build an FST which encompasses more ANGIE structure. In turn, the ability to recover some of the parse may be advantageous. Chapter 6 documents an entirely different approach to generating an FST. This is a structure which portrays each unique ANGIE column as a state or node on the automaton. Transitions on the FST are representative of transitions from one column to another. This approach directly takes account of the internal parse structure. Chapter 6 relates an algorithm which approximates ANGIE by compiling bigram probability estimates on these columns produced from ANGIE. The bigram probabilities are recorded on the FST arc weights, and symbols from the preterminal or phoneme layer are selected as output. This method is considerably more flexible in its ability to support unseen data, and becomes our method of choice for the final flexible vocabulary system.

3.7.3 Additional Comments

The technology to generate an FST from ANGIE eventually empowers us to efficiently augment the first-stage recognizer with sublexical modeling. For one, effective integration of ANGIE in stage one substantially enhances the power of the language models, directly translating to performance improvement. And using an FST affords much savings in computation and speed. However, the ANGIE-FST does not render the dynamic parser obsolete. First, the dynamic parser is essential for training the probabilities of the ANGIE models. Furthermore, the ANGIE parser is employed during the second stage, as another constraint. Here, we trade-off a significantly pruned-down search space for the full use of ANGIE's rich models that can generate multiple alternative parses for an unknown sequence. The dynamic parse mechanism seems more appropriate here, as part of its role is to provide a best guess at the complete sublexical structure of the unknown word.

3.8 Incorporating Grapheme Information

Let us briefly introduce our final major design consideration: the utilization of grapheme information. Two factors drive us to explore this:

1. Grapheme information may offer valuable low-level constraints over and above the existing language models. This may additionally reinforce the first stage. For example, the letter sequence *chr* appear exclusively at the onset of a syllable and the letter sequence *pt* is diallowed at the syllable onset in English.

2. This could facilitate access to spelling hypotheses of any unknown words directly upon their detection. We particularly favor this scheme because we argue for unknown word detection and transcription within the single framework.

There have been few previous attempts to incorporate graphemics into recognition, as this generally undermines recognition performance. But the concept of learning large numbers of new words from spoken input with the assistance of instantaneous sound-to-letter conversion remains attractive. At this point, some of the necessary mechanisms appear to be in place for fulfilling this goal:

1. ANGIE is suitably structured to model multiple sources of knowledge residing below the word level. In fact, grapheme units have been incorporated at the terminal layer in pre-

vious implementations for the purpose of bi-directional letter/sound conversion [69]. Although such a solution precludes the simultaneous modeling of pronunciation and grapheme rules, an alternative method will be presented.

2. The FST conducts string mappings from one alphabet to another. Hypothetically, the FST could be configured such that providing a phonetic sequence input leads to a set of letter hypotheses for an unknown word, emitted at the output.

3. Once constructing an ANGIE-FST becomes a viable path, we can envision a data structure that produces probabilities expressing constraints that include spelling ones. This is collapsed into a compact and efficient format, and the spelling hypotheses are easily accessible.

The key strategy is the invention of a new set of units for modeling purposes, to be placed at the pre-terminal layer in the ANGIE parse tree. These are termed *"letter-phonemes"* due to their dual purpose for capturing spelling and pronunciation. We expand on their characteristics in Chapter 6 and also assess their utility. Essentially, the ramifications of this modeling device are the following:

- There now exists a larger phoneme-level symbol set embedding more linguistic context, leading to tighter constraints and lower perplexity measurements. Any interactions or phenomena related to graphemics and phonological processes are modeled in the lower layers of the ANGIE tree.
- These symbols can be combined in novel ways to form new words where their spellings and pronunciations are clearly defined. The latter feature will aid us in making hypotheses corresponding with unknown word sequences.
- As introduced in Section 2.4.1, our lexical units, called *morphs*, already merge pronunciation and spelling information together in a syllable-sized unit. It will be apparent soon that representing these morphs in terms of *letter-phonemes* is a natural strategy. These two modes of representations are well-matched, and letter-phonemes in conjunction with intermediate morphs give intuitive representations for new words.

3.9 Final Remarks

Over the last three chapters, we have thoroughly covered (1) the research challenges along with the previous attempts addressing them, and (2) the visions behind our strategy, their motivations and underlying ideas. The next chapters of our thesis detail a series of experiments which will validate each of these novel concepts. Each experiment will build upon the findings of the previous one, and establish the feasibility of a novel feature in contributing towards a near-real-time flexible vocabulary speech understanding system.

Chapter 4

A Preliminary Two-Stage System

4.1 Overview

This chapter traces the implementation and evaluation of a preliminary system. We study the feasibility of ideas that will eventually form some of the fundamental building blocks for our envisioned flexible vocabulary system. We begin by defining the motivating factors. This is followed by a description of the two-stage architecture, along with the various modeling components employed in each stage. Next, a set of recognition experiments on JUPITER-based (in-vocabulary) sentences is presented. The chapter concludes with an analysis of the findings and their consequences, and suggests the possible directions from here on.

4.2 Motivation

Our first experiment assembles a simple two-stage system to test some basic hypotheses that were first proposed in Chapter 2. An initial two-stage system has been designed to embody the following ideas:

- The first stage prunes the search space significantly through exerting powerful acoustic and tightly constrained language models.
- The interface between the first and second stages is a network containing the hypotheses of the first stage reduced back to their phonetic sequences.
- The second stage attempts to tightly couple together the application of several higher-level knowledge sources.

Part of our endeavor is (1) to demonstrate the possible gains harnessed from a tightly coupled integration of disparate knowledge sources that are derived from rich hierarchical language models; and (2) to show that this can be effectively implemented by a two-stage solution with a phonetic network interface. We posit that reducing the hypotheses back to the individual phones constitutes a more compact way of representing the search space.

Another critical point of interest is the effect of relaxing linguistic constraint at the first stage. This will be an investigation on the possible adverse impact of departing from word-level knowledge at the initial pass, and whether the potential information loss can be recovered using an effective second stage. This will be accomplished by breaking away from the word lexicon in the first stage and using a less constrained morph lexicon. We shall draw comparisons in our experiments.

Furthermore, the system design will entail combining the major frameworks of ANGIE and TINA[1] with a segment-based speech recognizer. It has long been envisioned that, given a tightly coupled integration scheme, ANGIE and TINA can form powerful language models by operating in concert.

Although this simple implementation is conducted in the JUPITER domain only, the two-stage paradigm is intended to be an initial step towards separating the application of information that can be considered as domain-independent or domain-specific. Further steps will need to be taken later directing the first stage towards more domain independence.

4.3 System Architecture

Figure 4-1 displays the initial configuration for a two-stage system. The first stage consists of a segment-based recognizer which employs context-dependent acoustic models. This first-stage core recognition engine is based on the SUMMIT [28] system developed for the JUPITER weather information domain. The relative performance of using a word-level lexicon against an alternative lexicon using *morph* units is assessed here in this stage. Both lexicons are supported by n-gram language models on the respective units. The output of the first stage is an N-best list of word hypotheses. These are decomposed back to the original phonetic sequences from which a very compact acoustic-phonetic network is created. The second stage loads this network, and conducts a search coordinating the joint application of multiple

[1] Further explanations on TINA, the natural language module, is provided in Section 4.6.2.

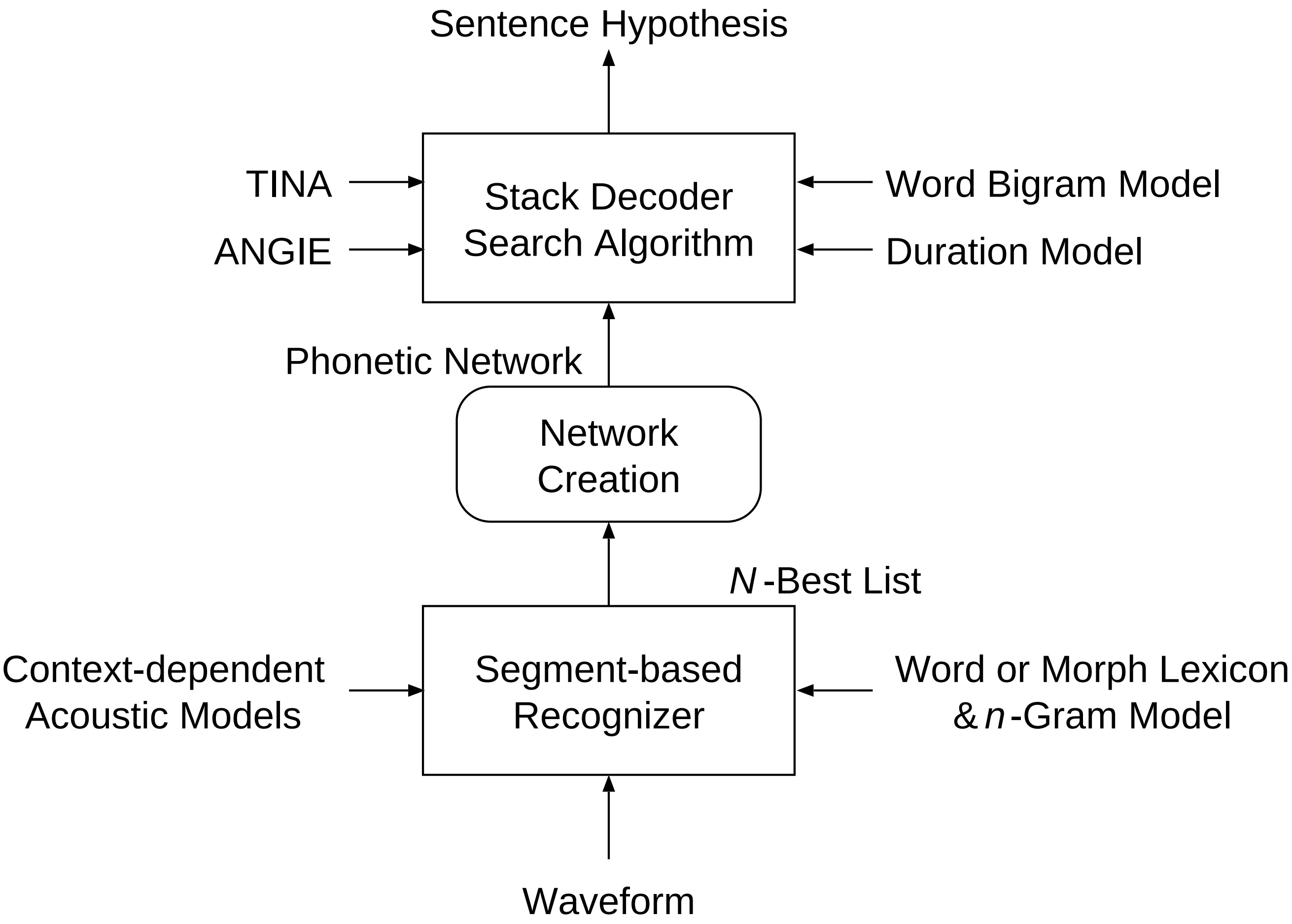

Figure 4-1: *Block Diagram of the Preliminary Two-Stage System.*

knowledge sources. This primarily consists of the ANGIE and TINA frameworks together with a word bigram model and an ANGIE-based duration model [11]. The second-stage search ultimately yields the final sentence hypotheses. The following sections will elaborate on the details of both stages as well as the intermediate network creation procedure. These include the lexicon, acoustic and language models of the first stage, and the ANGIE and TINA integration process in the second stage.

4.4 Stage One

4.4.1 The Lexicon: Words versus Morphs

In the experiments of this chapter, part of our intention is to discover the degradation in performance affected by partially stripping away word-level information in the first stage, shifting word and above word-level modeling entirely towards the second stage. In Section 2.4, we have weighed the various alternative choices of lexical representation available from the linguistic hierarchy. It has been determined that a unit at the syllable layer has many desirable properties for our modeling purposes. Section 2.4.1 first introduced the concept of a morph lexicon. We have argued that using these morphs imposes the necessary phonotactic constraints and also enhances predictive power by adding contextual knowledge associated with spelling and position within the word. At this preliminary phase, it must be noted that these morph units retain a high degree of domain dependence. This is due in part to the fact that (1) rich linguistic context is encoded and (2) experiments are strictly undertaken in the JUPITER domain where the total number of morphs is limited.

A measure designed to better capture long-distance constraints is the use of multi-word units. Words which commonly co-occur in adjacency are assigned together as a single word. They are represented by the component words connected with an underscore. These include word pairs in common city names such as *New_York*. The subset of function words that contain multiple components is displayed in Table 4-1 below.

In the word lexicon, these were originally included to (1) improve perplexity for the language model and (2) enable alternative pronunciations specific to these word pairs[2]. We have simply retained the function word subset in the morph experiments for reinforcing

[2] Note that these units were empirically determined, rather than via some automatic process. They have evolved during the fine-tuning and development of the JUPITER system.

constraints in the morph lexicon. This subset is a list of 19 compound function words. Certainly, as Table 4-1 reveals, these multi-word units commonly arise, and we expect that collapsing them into single entities in the morph lexicon will greatly boost the language model. This can be justified though, as many of these words do not pertain specifically to the topic domain, and could conceivably be kept as part of any domain-independent lexicon.

can_you
do_you
give_me
going_to
how_about
i_am
it_will
i_will
i_would
it_is
thank_you
that_will
what_about
what_are
what_is
what_will
when_is
where_is
you_are

Table 4-1: Set of Compound Function Words in the JUPITER Domain.

All the lexical units are given by their respective phonetic baseforms which allow for alternate pronunciations. The mechanism for modeling pronunciation variations among lexical units is the pronunciation network that is generated from the list of baseforms [109]. During this process, a set of hand-written phonological rules are successively applied in sequence to each word on the network. As part of the SUMMIT recognizer, this has been the conventional approach to pronunciation modeling. Such a rudimentary method suffers from some major drawbacks:

- For each word, the alternate pronunciations have to be determined individually. None of the rules can be shared among the words or generalized to apply to new words.
- The system of rules require sequential application, and are hand-crafted in such a way that adding or altering rules becomes a complex procedure.
- The rules are strictly applied on the basis of local phonetic context, and fail to observe a longer distance effect.
- The pronunciation network does not utilize probability modeling.

In spite of these obvious shortcomings, the current set of initial experiments retain this approach for simplicity while we focus on other issues. Improved phonological modeling in the first stage will be addressed in Chapter 5.

4.4.2 Acoustic Modeling

Our segment-based recognizer, SUMMIT, performs segmentation by postulating boundaries between phonetic segments at *"landmark"* regions wherever the rate of change in the spectral features reaches a local maximum. A network of possible boundaries forms the segmentation lattice. For acoustic models, it utilizes context-dependent diphone boundary models. These are categorized into two types: internal or transitional. Both types are trained from examples of boundaries proposed by the segmenter during training. The internal ones correspond with boundaries that are not at endpoints of actual phonetic segments; that is, they are regions of acoustic change within the same phone. A transitional model is trained on the actual boundaries between phonetic segments. These models capture coarticulatory effects from speaking one phone to the next. Hence, modeling a particular phone segment will require a number of internal models as well as the boundary models that capture the left and right contexts at the endpoints of the segment.

In total, there are 68 phonetic units. Due to the lack of sufficient amounts of data, many of the acoustically similar diphone units are collapsed together into equivalence classes, pooling the data to form a single model. The combined set of transitional and internal units total 631 models. For acoustic features, 14 Mel-scale cepstral coefficient (MFCC) measurements are computed at every 5 msec frame interval. Subsequently, 8 different averages of these MFCCs are taken from regions within a 150 msec window surrounding each boundary. Using varying length time windows on both sides of the boundary serves to capture more acoustic context. The result is a 112-dimensional vector. Principal components analysis is used for reducing the number of dimensions to a total of 50. The classifier uses diagonal Gaussian mixtures with a maximum of 50 Gaussian kernels.

4.4.3 Language Modeling and Search

For language modeling, the system incorporates a bigram in a forward Viterbi search. This yields a Viterbi lattice and the best scoring hypothesis. An A^* search is applied backwards on this Viterbi lattice, extending the path hypotheses by one word at a time. An A^* search is a best-first search that increments the score with a look-ahead estimate that examines the path from the current node to the end. In the backward search, the estimate refers to a path from the current node to the beginning in time. This estimate must be an upper-bound, or an over-estimate for the search to be admissible. The score from the forward Viterbi search, at each Viterbi node is the best partial path so far from the beginning in time, and is therefore the highest score. This constitutes the over-estimate required for the A^* search. In cases where two paths arrive with the same word sequence at the same boundary, the inferior path is pruned away. Because the Viterbi search itself is ill-suited for applying longer distance models, the backwards A^* search is the mechanism by which a reversed trigram is applied. Ultimately, the N highest scoring sentences are generated at the output. For the word recognizer, the bigram and trigram models use equivalence classes to improve perplexities. The equivalence classes number at around 200.

4.5 Network Creation

The first stage outputs a small N-best list. From this, the phonetic sequences of the candidate words are retrieved. The algorithm performs a lookup for the associated phonetic segment and recovers the context-dependent acoustic scores. Subsequently, an acoustic score for each segment is composed by accruing the left-context transitional and internal diphone acoustic scores for a segment. Whenever a segment has multiple candidates for left phonetic contexts, the context producing the highest score will be chosen to approximate a context-independent acoustic score. The next step entails collapsing segments with start and end points that coincide in time. Nodes on the network denote time boundaries, and every arc is linked with a phone candidate and its associated acoustic score. Note that the first-stage language model scores have been excluded entirely.

This generation phase is a heuristic means for constructing phonetic networks in a post-processor to stage one, applied without altering the existing components of the SUMMIT recognizer. It allows us to probe the feasibility of the phonetic network in an initial effort. Besides, ensuring a short N-best list will avoid heavy or redundant demands on computation imposed by creating the network.

It is possible that converting the N-best list to a network will engender a *cross-pollination* effect. Under such circumstances, even candidates not contained in the original N-best list can be hypothesized by the second stage. This is due to an expansion in search space when phonetic segments extracted from the N-best list are collapsed together and connected.

4.6 Stage Two

4.6.1 The ANGIE Parse Mechanism

The original ANGIE parse mechanism is employed here within the integrated search. As mentioned in Chapter 3, the lexicon is configured into two tiers by a word and a morph lexicon. During the second-stage parsing, the operation is expedited by imposing constraints from the ANGIE morph lexicon. It should be noted that the morph lexicon is identical to that used in the stage-one morph experiments. Hence, the phonetic hypotheses passed from the first to the second stage originated from the same morph inventory, and are well-matched to potential ANGIE parses here.

There are approximately 100 "pseudo-phoneme" categories, 40 categories in the syllabification layer and 8 categories in the morphology layer. The JUPITER grammar also includes a large set of multi-word units which includes the subset stipulated by the first-stage lexicon. The rules of ANGIE have been engineered to treat these designated words as multi-syllabic single words. This is justified because these clusters (generally pairs) of words are frequently uttered together.

One ramification is that some effects of sentential stress are captured as lexical stress in ANGIE. For instance, *new york* is designated as a single word with lexical stress bestowed on *york* and *new* is assigned as an (unstressed) prefix morpheme. For compound words such as *what_is* and *what_will*, ANGIE subsumes the respective contractions, *what's* and *what'll*, under the one compound unit. And these contractions are modeled as alternate pronunciations, contributing to the statistical models that describe sublexical patterns. This is an artificial way of encapsulating some word-level information within ANGIE's models.

4.6.2 The TINA Framework

TINA is a natural language system developed for spoken language applications, first introduced in [95]. It models natural language using a framework that shares a number of common features with ANGIE. First of all, the model forms a hierarchical parse tree based on a context-free grammar, defined by hand-written rules. Underlying the framework is a statistical model that is trained by parsing a set of training sentences. Just as in ANGIE, probabilities learned during the training phase capture knowledge in addition to the constraints which have been engineered by the hand-written rules. Conditional probabilities are computed on the context of internal tree nodes, rather than the production of the associated rules. More specifically, within a parse tree, the probability of a node depends on sibling-to-sibling transitions conditioned on the parent node context. This is the top-down equivalent of the philosophy adopted by ANGIE. Consequently, the probability scores are conveniently designed to predict the next-word candidates, given the previous word hypotheses for the current sentence. One advantage is that a recognizer can easily access an NL score for the next probable word. TINA implements a top-down control strategy, and is augmented with: (1) a set of features that enforce both syntactic and semantic constraints, including number and verb tense agreement, and (2) a trace mechanism used to handle movement phenomena. The latter refers to the incidence of gaps commonly associated with *wh*-queries in English.

TINA has been developed especially to cater to processing speech recognizer outputs which are riddled with recognition errors and artifacts of spontaneous speech such as agrammatical constructions and speech disfluencies. These would generally lead to parse failures. However, in the event of a failure to construct a full parse tree, the *robust parse mechanism* [94] takes over, and retains a partial parse that carries the admissible sentence fragment. In this way, a partial meaning representation can be generated.

With the abovementioned list of attributes, TINA has long been envisioned to aid recognition both as an NL processor and a linguistically motivated language model endowed with long-distance constraints. When faced with an unknown word, TINA can be configured to support its occurrence under a number of specified categories such as a proper name[3]. At the same time, a meaning representation can be generated. This affords significantly more functionality when compared with traditional n-gram models. The latter are ill-equipped for supporting unknown words, are difficult to update and deficient at capturing long-distance information.

A major obstacle that remains then is the direct integration of TINA into the active search process. The goal is to couple TINA closely with the acoustics-driven search so that as early as possible, exploration of paths can be perturbed on the basis of higher-level information. Some previous work has investigated this issue using TINA. In [98], TINA was employed in an A^* search over a word network of hypotheses, parsing the candidate sentences and re-ranking them for the best scoring one afterwards. Our work here investigates the use of

[3] This will be put to use in our experiments. See Chapter 8.

TINA integrated with ANGIE. It is ambitious in its attempt to apply information from above and below the word level in a control strategy that allows them to mutually interact. Our hope is that the quality of the output is enhanced under this tightly coupled environment. The compromise will be the need for a small phonetic network at the input in order to keep the search computation tractable.

4.6.3 The Integrated ANGIE-TINA System

Central to the second stage is the control strategy that coordinates the application of multiple word-level knowledge sources along with ANGIE under a single search. As reiterated above, the primary challenge is to tightly couple together the disparate sources of linguistic information. We choose to configure the search so that it begins in a bottom-up manner and proceeds to apply high-level models. By this, we mean that the search is guided by the phonetic hypotheses of the first stage, and those hypotheses are processed by ANGIE. When ANGIE discovers a potential word candidate along a phonetic sequence, the putative word is processed by subsequent models, namely, the NL analysis via TINA, word bigram and duration model. This strategy necessitates an algorithm that monitors large numbers of possible paths, and ANGIE and TINA each individually maintain large numbers of partial theories. Naturally, this calls for large computational and storage requirements. Our solution is to begin with a small phonetic network as input and to utilize the *stack decoder*.

The stack decoder is a one-pass, left-to-right algorithm, described in [44, 80, 82, 81]. Our version of the algorithm is illustrated by the diagram in Figure 4-2, and was first employed in Lau's work [59]. A data structure maintains a stack of all partial paths. These are sorted by increasing order of time and then by decreasing order of score. The path, pending for the next removal from the stack, is always at a boundary that is the furthest behind in time, where the boundary is yet to be completed. This path is the highest-scoring one at that boundary. A new phone candidate, dictated by the phonetic network, extends the current path that has been popped from the stack. (Refer to step [1] in diagram.) The top-level procedure will intermittently consult various modules, and maintain the total score of the extended path. Stacks of ANGIE and TINA partial parses corresponding to this path are also stored. First, the ANGIE module will advance the corresponding partial parses by the new phone. If ANGIE succeeds in proposing a parse, the corresponding linguistic score is returned and incremented to the total score. Additionally, if a word ending is possible, the

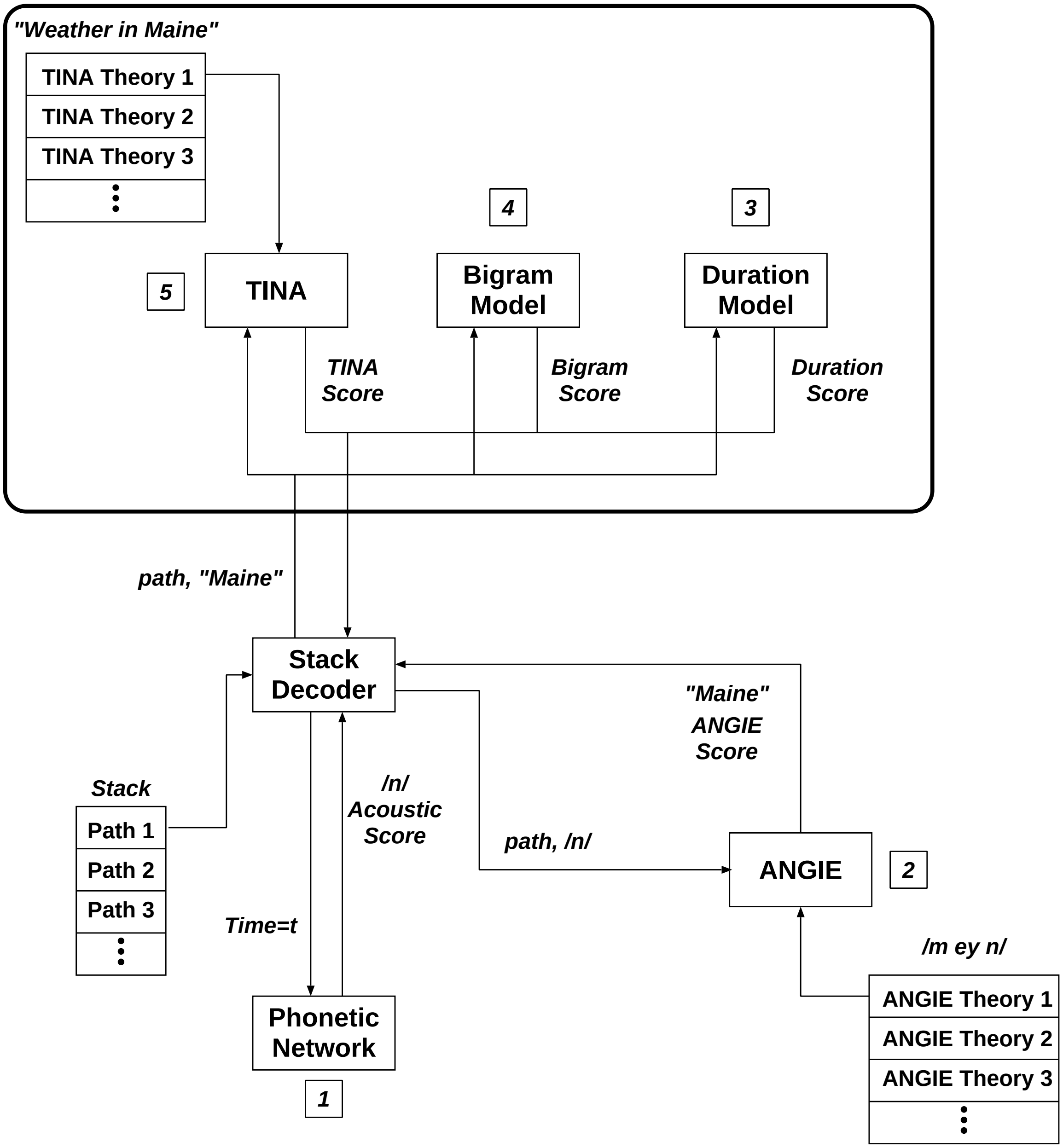

Figure 4-2: *Illustration of the* ANGIE-TINA *Integration Strategy. See Section 4.6.3 for detailed explanation.*

word candidate is also returned. (Refer to step [2] in diagram.) At this point, the decoder retrieves a duration and bigram score for this hypothesized word. (See steps [3] and [4] in diagram.) Finally, it will enlist the TINA parser. Each of the TINA partial parse candidates are extended by the new word. (See step [5] in diagram.) The path may be eliminated if failure is encountered for every possible parse. Otherwise, an NL score is returned to the top-level decoder, and the total score is re-computed. To further ease computational load, the robust parse handling within TINA has been shifted from the parser into the recognizer search. Along a given partial path, TINA will attempt to parse, from the left, as many words as possible, only using full parse theories. When a failure occurs, the algorithm then backtracks leftwards to the first word where a sentence ending is permitted, and where the right adjacent word can begin a new sentence. The parsing will proceed forward from then on. This heuristic method will improve speed by restricting TINA's internal search to support only full parse theories. Pruning is conducted at multiple levels by each module individually. For stack pruning, a fixed constant n number of paths are allowed for a given time boundary. In our implementation, the application of TINA models is optional. This will allow us to switch the integrated NL component on or off for comparisons during experimentation.

The stack decoder has the advantage that all theories which end at a particular point in time are explored together as a group. Thus, the theories competing against each other during the search process all cover the exact same acoustic space. This is opposed to previous work by Lau [59] where the search procedure extended paths of varying lengths in time. To ensure fair comparisons among scores, elaborate normalization constants need to be factored in. Also, tighter coupling occurs because scores are unified at any proposed word boundaries, while the search progresses from left to right. This is distinctly different from word graphs where all possible word hypotheses are generated for the entire sentence prior to the application of external information. Finally, it is a future goal to also generate meaning representation directly at each step during the search but, at this juncture, it is beyond the scope of our work.

4.7 Recognition Experiments

4.7.1 Experimental Method

This JUPITER implementation utilizes a 1341-sized word lexicon. There are 1603 items in the morph lexicon. The first stage is set to produce an N-best list with $N = 10$. The ANGIE grammar and the hierarchical duration model are both trained on 11677 utterances. These were forced alignments obtained *a priori*, seeded from a SUMMIT recognizer. In the training data, 62.4% of the words are fully specified as a single unit in the morph lexicon either because they are monosyllabic or they are part of the subset of multi-word units. The TINA word grammar is separately trained on 6531 utterances. All experiments are evaluated on an independent test set of 352 utterances.

For baseline comparison, we use a single-stage SUMMIT system, which outputs an N-best list (with $N = 10$). At the time of undertaking this experiment, this baseline system was competitive with the state-of-the-art system being deployed for JUPITER. It is also identical to the first stage of the two-stage architecture, with the same context-dependent acoustic models. We enlist the word-level lexicon, along with the bigram and trigram language models.

We present two variations of the baseline system:

- **System 1 SUMMIT Top 1:** The best scoring sentence candidate is chosen.
- **System 2 SUMMIT N-Best:** An elementary algorithm processes the N-best list to select from one of the N sentences. By applying TINA as a post-processor, the procedure will seek the highest-ranked sentence that satisfies a full parse. Failing that, it will repeat on the entire N-best list using the robust parse mode. If again this does not succeed, a parse failure has resulted, and none of the sentences are selected. This method does not incorporate the NL scores computed by TINA but prefers the most likely sentence where a meaning representation can be extracted. If parse failure ensues, no meaning representation is produced. This reflects that the understanding component cannot make sense of the recognizer output. The SUMMIT N-best mode was used by the real-time JUPITER system at the time this experiment was undertaken.

We report on the performance gains derived from using the two-stage system with ANGIE only in System 3, and the fully deployed ANGIE-TINA in System 4. Systems $3m$ and $4m$

Sentence	Key-Value Pair
Yeah, can I find out what the temperature is going to be like in Boston Massachusetts tomorrow morning	WEATHER: temperature CITY: Boston REGION: Massachusetts DATE: tomorrow TIME_OF_DAY: morning
What is the average rainfall in June	QUANTIFIER: average WEATHER: rain DATE: June
Help, what do you know about tornado advisories in Kansas	CLAUSE: help WEATHER: crisis CRISIS_TYPE: tornado REGION: Kansas
That's all, thanks	CLAUSE: close off
What do you know besides weather	CLAUSE: help BESIDES: weather

Table 4-2: *Typical Examples of Utterances and their Key-Value Pairs Used in the Understanding Evaluation.*

refer to the two-stage systems with the word lexicon replaced by morphs in stage one. For further examination, the first stage morph recognizer is isolated from the two-stage system. This is referred to as System $1m$. It is equivalent to a single-stage SUMMIT system using a morph lexicon and morph-based language models.

4.7.2 Understanding Evaluation

Before proceeding to present the experimental results, we highlight the importance of an appropriate evaluation method. In dealing with a speech understanding system, it is natural to look beyond word error rate. Ultimately, the goal is to maximize the completion for each task, and to achieve user satisfaction. At present, we move closer in this direction by attempting to quantify understanding rate or concept accuracy. We feel this is critical because an improvement in the system's ability to comprehend spoken input may not be entirely reflected in the raw word recognition accuracy.

We have devised an evaluation measure which captures the salient points of meaning from each sentence, discounting the less significant recognition word errors in the process. This is based on the semantic representation, derived from the TINA module. Given a recognition hypothesis as input, TINA generates a parse tree which can be automatically translated to a semantic frame representation. From this, we employ GENESIS [29], a language generation module, to paraphrase the frame into a set of pre-defined key-value pairs. These pairs are empirically determined by judging which information in the semantic frame is important in completing the JUPITER inquiry. As a result, the semantic frame is transformed to a simpler, collapsed meaning representation that encompasses only the essential information required to process the query. Examples of key-value pairs are given in Table 4-2.

To compute the final understanding error of a test set, we precompute the key-value pairs corresponding to the original orthographies of the set as reference. In cases of parse failures in TINA, this may be due to TINA's incomplete coverage, in which case the transcription is manually rephrased such that a parse can be generated while preserving the original meaning. In other cases, this may not be possible, because a percentage of the utterances lie outside the domain; that is, the spoken requests cannot be handled by the system, and no alternative phrasing would be interpretable by the dialog component. These reference key-values are deemed missing. The final understanding or concept error rate is a percentage calculated from the total number of mismatches, deletions and insertions against the reference key-values:

$$Understanding\ Error\ Rate\ (\%) = 100 * \frac{Subs + Dels + Ins}{Chances} \tag{4.1}$$

For missing key-values, in either the reference or hypotheses, deletions are counted.

Because this TINA evaluation module is identical to the NL module deployed in the recognizer, we feel that the evaluation method is a fair reflection of overall understanding performance. It simulates the situation where the system in evaluation is integrated with the dialog module, and will reveal whether the recognizer sentence outputs contain the important words required for extracting meaning. Utterances with mismatched key-values would be interpreted erroneously by the dialog component of the real system; that is, a wrongful action would result.

4.7.3 Results and Analysis

We will begin by reporting results for the integration experiments using the word-based first-stage recognizer, followed by results for the morph experiments. Recognition and understanding errors for the Systems 1–4, mentioned above are reported in Table 4-3.

System	Word Error Rate (%)	Understanding Error Rate (%)
1. SUMMIT Top 1	12.3	19.4
2. SUMMIT N-Best	13.4	17.0
3. ANGIE only	10.4	16.2
4. ANGIE-TINA	11.1	14.1

Table 4-3: *Recognition and Understanding Performance. Errors are given for systems described in Section 4.7.1. Systems 1 and 2 represent single-stage baselines, and Systems 3 and 4 represent two-stage systems with word lexicons at the first stage.*

When ANGIE is applied in System 3, the WER reduces by 15.4% (from 12.3% to 10.4%) compared with the baseline System 1. This system, without the aid of any NL knowledge, achieves an UER of 16.2% compared with 17.0% achieved by System 2, which employs NL processing. When ANGIE-TINA is fully deployed in System 4, 11.1% is achieved on WER and 14.1% is achieved on UER. These outperform both Systems 1 and 2. In particular, this translates to a 17.1% relative reduction in UER compared with System 2, which also employs NL.

It is clear from the results for System 3 that performance benefits significantly from the combined probabilistic sublexical models of ANGIE and its duration model. While the baseline system lacks any kind of statistical pronunciation model, the ANGIE framework applies probabilistic reasoning to both word-internal sublexical phenomena as well as cross-word phonological and coarticulatory effects. The latter is due in part to the inclusion of multi-word units that observe effects between adjacent words. These factors alone have contributed to a superior understanding performance prior to introducing NL information.

Secondly, an integrated ANGIE-TINA in System 4 has exhibited superior understanding performance compared with the more conventional N-best list post-processing in System 2. As we postulated, an integrated search strategy enables more meaningful partial paths to proceed.

When N is raised to 100, the ANGIE System 3 does not improve significantly, although for the ANGIE-TINA System 4, the understanding error improves to 13.6%. We can conclude that the ANGIE-TINA guided search retrieves a greater number of correct paths from the deeper network.

Final results for the morph-based experiments are tabulated in Table 4-4. It can be observed that the WER of 11.8% in System $3m$ outperforms that of System 1 with 12.3%.

System	Word Error Rate (%)	Understanding Error Rate (%)
$3m$. ANGIE only	11.8	18.1
$4m$. ANGIE-TINA	13.9	17.3

Table 4-4: *Recognition and Understanding Performance for Two-Stage Systems. These use morph lexicons instead of word lexicons at the first stage.*

Similarly, for System $4m$, understanding performance of ANGIE-TINA, at 17.3%, is comparable to that of System 2 at 17.0%. From this, we infer that the sophisticated language models of ANGIE and ANGIE-TINA recover most of the loss in performance in stage one incurred by the morph lexicon.

It is noteworthy that in comparing the word-based systems against their morph-based counterparts, the former utilize a word trigram at an early phase, in the first pass, whereas the latter do not employ this information. Instead, at the word-level, they rely solely on the bigram and ANGIE-TINA, applied only in stage two. We claim that performance would further improve if a word trigram were incorporated.

An additional caveat is the observation that WER does not necessarily fall in accordance with UER. This is apparent when we compare System 1 with System 2, System 3 with System 4 and System $3m$ with System $4m$. In fact, a slight rise in WER emerges upon imposing NL constraints for the following reason. Under some circumstances, the NL module favors a candidate hypothesis which involves inserting an additional word error. This hypothesis associates with a lower recognition score (derived from the acoustic and n-gram language models alone). But this ultimately results in a more likely parse and the correct intended meaning. The implication is that some word accuracy may need to be sacrificed for the sake of improving understanding rate when considering the underlying objective of extracting the true meaning or intent behind the spoken input.

We gauge the drop in performance from switching to morphs in stage one by computing the morph error rate (MER). This is calculated in the same way as WER but at the level of the morph unit. Equation 4.2 below defines MER.

$$Morph\ Error\ Rate\ (\%) = 100 * \frac{Substitution + Insertions + Deletions}{Total\ Number\ of\ Morphs} \tag{4.2}$$

From the single-stage morph recognizer, System $1m$, the highest scoring hypothesis is se-

System	Morph Error Rate (%)
1. SUMMIT Top 1	10.8
1m. Morph SUMMIT Top 1	12.8
3m. ANGIE only	10.9

Table 4-5: *Comparison of Morph Error Rates for Selected Systems. See text for explanation.*

lected. This is compared with (1) the morph decomposition of the best-scoring output of the baseline word recognizer, and (2) the morph decomposition of the sentence output of System 3m, using ANGIE only in stage two. Results are shown in Table 4-5. There is an 18.5% deterioration in MER (from 10.8% to 12.8% error) comparing the morph output of System 1m with System 1. As these recognizers are identical except for the lexicon and n-gram models, the degradation has directly resulted from relaxation of language constraints. But MER falls to 10.9% for the morph output of System 3m, the two-stage ANGIE system. That is, at the morph level, errors incurred by using morph units only in stage one have largely been recovered by the ANGIE-only second stage.

Although we did not conduct comparisons on the basis of computation, note that all systems operate within near-real-time. Minimal effort was devoted to optimizing the computational speed, although it is a primary endeavor to take approaches that impose manageable computational demands.

4.8 Discussion and Summary

These experiments have contributed to some encouraging results for an initial investigation. We are led to conclude the following points:

- We have succeeded in incorporating NL, phonological and durational constraints under a tightly coupled control strategy. By limiting an explosion in the size of the search, it has been possible to apply the rich hierarchical linguistic knowledge in concert, allowing the mutual interaction and feedback. This has significantly enhanced performance.
- The phonetic network has been an effective means of interfacing the two stages. This stands despite the fact that it was formed by a heuristic method from a very small N-best list. The network contains only context-independent acoustic scores that are

computed by an approximation, and *all* language model scores from the first stage have been omitted. Furthermore, returning to the more basic atomic units enables the second stage to propose an entirely new set of words. These are not tied to the word or morph hypotheses selected by the first stage.

- Consequently, the second stage is a natural phase for applying the ANGIE parse mechanism over a small pruned-down set of phones. The results indicate that ANGIE's powerful probabilistic sublexical framework accounts for much error reduction over a system with no probabilistic phonological modeling.
- At this preliminary phase, our attempt to distinguish between low-level and high-level information under a two-stage paradigm has proven to be viable. The performance has been comparable with a state-of-the-art system. We attribute the positive results to the predictive power of the morph units and their respective language models. These encode the necessary syllable-level information as well as other valuable linguistic context. However, under the narrow topic domain of JUPITER, there are in fact a greater number of morphs than words. But for very large corpora, the number of morphs will eventually be dwarfed by the number of words they map to. In any case, a much larger set of words can be generated from the current set of morphs in use. This sort of generality will be advantageous when we encounter OOV words.

This chapter has dealt with an initial experiment but has served to successfully validate some of our fundamental ideas. There is an array of issues associated with this current system that we will proceed to tackle. First of all, the first stage remains far from being domain independent. The morph units are intentionally designed to preserve as much contextual information as possible, and as a consequence, they are inextricably intertwined with the topic domain. We have yet to show that a more general set of units could possibly afford comparable predictive power. By the same token, the high performance of the two-stage architecture is linked with a heavy reliance on contextual knowledge used in the first stage. When morphs are used, the morph trigram exerts a large degree of linguistic constraint. This system would be ill-equipped to handle unknown words because the powerful n-gram constraints, exercised by the first stage, are centered on the fixed vocabulary of morphs. Therefore, we will address the issue of introducing more flexibility and generality in the upcoming chapters. To do so, we ensure that more domain-independent yet constraining knowledge is allotted in the first stage. This will begin by the integration of ANGIE models in the first-stage recognizer, examined next in Chapter 5.

Chapter 5

A Finite-State Transducer Based System

5.1 Overview

This chapter introduces some improvements from the two-stage architecture of Chapter 4. First of all, we recapitulate some of the main concerns of the previous two-stage system, and motivate the features of the new FST system. This is followed by a description of the system architecture. We give an overview of the re-implementation of the first stage under the FST paradigm. After this, we elaborate on our ANGIE-FST engine, the ANGIE grammar developed for the FST, and the method used to generate the FST structure. The main characteristics of the second stage will be briefly highlighted. This two-stage system is used in a set of recognition experiments. Finally, we present the results and discuss some implications for them.

5.2 Motivation

Now that a two-stage architecture with a phonetic network interface seems a feasible solution, let us focus on the remaining issues of the first stage. It has been seen that the previous first stage in Chapter 4 still relies to a large extent on domain-specific contextual information. This holds in spite of using a morph lexicon which embodies greater generality than words. As already stated in Section 2.4, resorting to the most generic units such as those at the phone or phoneme level would offer maximal domain-independent generality. But this entails a sacrifice on word accuracy, by confounding the recognizer with too many alternative hypotheses. Our strategy is to amend our first stage by gradually introducing more domain-independent low-level knowledge. At each step, we must ensure that performance is not compromised beyond acceptability.

With this strategy in mind, our next goal is set to augment the first stage with ANGIE sublexical probabilities. Doing so brings us increased linguistic constraint and enhanced predictive power, but it is also compatible with the objective of using only domain-independent knowledge. As explained in Chapter 3, the problem of integrating the ANGIE framework early into the recognition search has been formidable. Fortunately, we are facilitated here by (1) the use of a smaller set of units, namely the morphs, and (2), more importantly, the advancement of FST technology in our segment-based recognizer. Let us revisit the main advantages in adopting an FST framework:

1. FSTs supply us with a parimonious means of representation for partially capturing ANGIE's models among other knowledge sources. When multiple disparate knowledge sources are all represented as FSTs, their combination and application is transparent to the recognition search.

2. They allow the entire language search space to be compiled into a single data structure prior to recognition time. The FST is subsequently optimized via standard routines for efficient run-time operation.

The FST paradigm folds ANGIE seamlessly and efficiently into an existing segment-based recognizer. It turns out that this flexibility will be a great asset when more complexity is added to the representation. Our last point will become more apparent to the reader in later chapters.

5.3 System Architecture

The new two-stage architecture is a natural extension of the preliminary system introduced before. It is illustrated in Figure 5-1. Again, the first stage is a syllable-level recognizer. The recognition engine employs the same segment-based recognizer, SUMMIT. But this has been upgraded to use weighted FSTs to represent all the search constraints. A single pre-computed FST will embed the combined linguistic information. These include the ANGIE

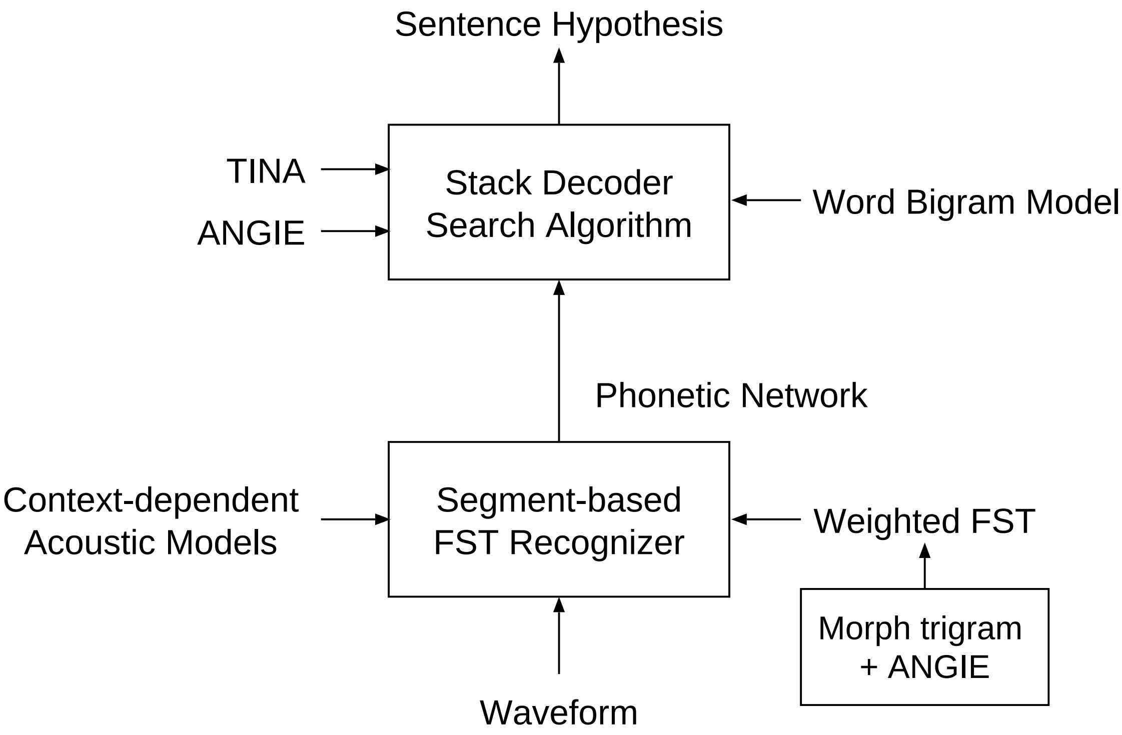

Figure 5-1: *Block Diagram of FST-Based Two-Stage System.*

sublexical probabilities and a trigram language model on a lexicon of morphs. This will be explained in Section 5.4. The first-stage morph hypotheses are reduced to their phonetic sequences, and an optimized phonetic network is automatically generated. Note that the network generation phase of the previous system has been eliminated. The second stage, described in Section 5.5, is similar to our previous work integrating ANGIE and TINA together in a single search, and outputs sentence hypotheses.

5.4 Stage One: The FST-Based System

The current first stage is the same segment-based recognizer used in the previous experiment. It has been modified to utilize weighted FSTs to define its search space. This work has been previously described in [110]. As first introduced in Chapter 3, the recognition task is now modeled in terms of an FST composition. We defined an FST, U, which models the entire search space via a series of transductions. A recognizer's role is to seek the highest-scoring path through a space covered by U. Conventionally, it is computed as:

$$U = C \circ P \circ L \circ G$$

where

- C maps context-dependent diphone labels of the acoustic model to context-independent labels,
- P applies phonological rules,
- L is the lexicon mapping pronunciations to words, and
- G is the language model.

In our work, one single FST is generated to transduce context-independent phone labels to morph labels. This is the role of the ANGIE-FST which embeds ANGIE probabilities in its arc weights. Thus, it functions to perform phonological and sublexical modeling along with lexical access, $P \circ L$, within a single step. For G, these experiments will consider both bigram and trigram models on the morphs. FSTs are easily generated from n-gram models from standard algorithms.

In striving to work under near-real-time conditions, we optimize on efficiency as much as possible. And so it is preferable to work with pre-composed FSTs whereby all the possible search paths and the linguistic scores are computed *a priori*. Greater efficiency is also achieved when a standard optimization is run beforehand. As remarked earlier, the routine involves determinization followed by minimization, which redistribute the scores without altering the overall search paths. These routines generally lead to more compactness. But, in our experience, caution must be taken to design FSTs that are actually determinizable[1]. Determinization allows arcs of the prefixes of sequences to be shared, thereby delaying the outputs. This may cause an initial expansion in the size of the FST. Minimization promotes sharing of arcs at the ends of sequences, and pushes outputs from the end towards the beginning. In cases where determinization causes an expansion in the FST size beyond that which can be handled by the computer, subsequent manipulation including minimization is no longer feasible. Alternatively, "on-the-fly" operation requires less memory but will slow down the recognition search process significantly. Hence, with the trade-offs between memory requirements and run-time speed, designing determinizable and compact pre-composed FSTs has been a principal consideration. The experiments in this chapter all deal with a system that uses the pre-computed FST U. This is pre-loaded into the recognizer prior to recognition.

[1] Discussions of determinization are available in the literature [71].

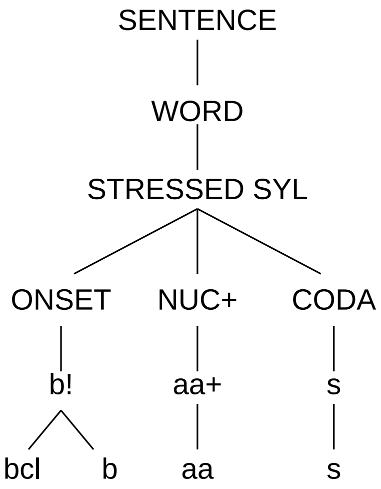

Figure 5-2: *Morph-Based ANGIE Grammar Parse Tree. This represents the morph bos+ for the word boston. The morphology layer simply categorizes the morph bos+ as a stressed syllable.*

The first stage outputs phonetic networks that conform with the FST format. They are optimized via the same minimization and determinization algorithms. These networks contain context-independent phone labels along with scores (derived from both the acoustic and language models) as the arc weights. These scores are added to the second stage scores with a weight which is empirically determined during development. The next sections will examine the ANGIE-FST in greater detail, beginning with the ANGIE grammar that is tailored for this system.

5.4.1 The ANGIE Grammar

The ANGIE grammar for the JUPITER domain has been re-designed to contain only up to the syllable-level. This is intended to create a leaner ANGIE grammar that would be more suitable for converting to a compact FST[2]. As the syllable-level first stage uses a morph lexicon that coincides with the ANGIE second-tier morph lexicon, a convenient way to prune down the ANGIE grammar is to discard information above the morph level, using only the single lexicon of morphs.

Figure 5-2 depicts a typical parse tree for this grammar. It represents a realization of the morph *bos+* which is manifested in the word *boston.* Notice that the morphology level, which previously encoded morphemic identity such as PREFIX, STRESSED ROOT, and so on,

[2] This also reduces dependence on the word lexicon.

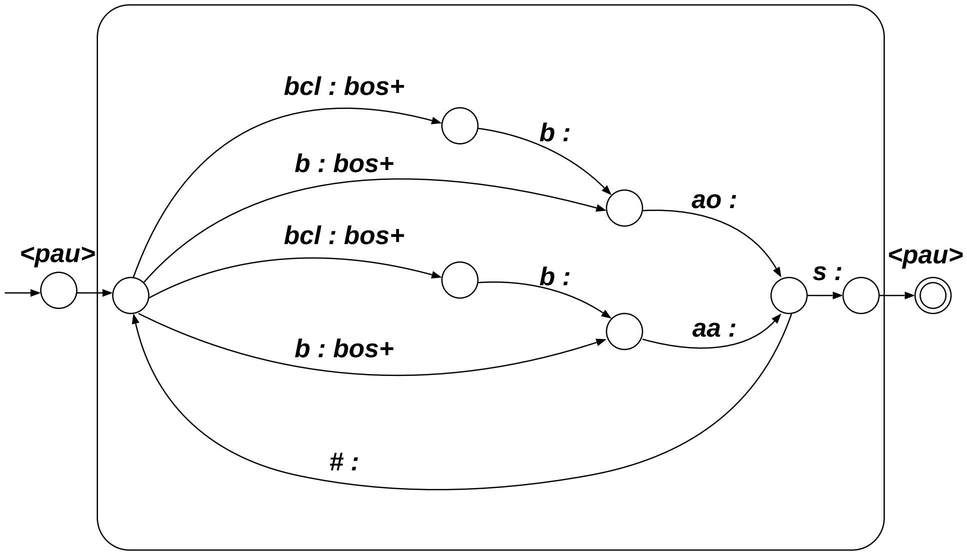

Figure 5-3: *A Simplified FST for the Morph "bos+" in Boston. Labels on the arcs denote the inputs and outputs, separated by a ":". Situated in the self-looping arc, a "#" is a special symbol denoting a morph boundary, as the arc returns to begin another phonetic sequence belonging to the next morph. The pause marker, "<pau>", marks the beginning and end of a sentence.*

has been altered. This now only distinguishes between stressed and unstressed syllable and consequently, knowledge of position within a word is omitted. We believe that this grammar preserves a large portion of the constraint of the original because the important knowledge such as the syllable phonotactics and phonological phenomena are retained. The loss in morphological information will be partially recovered with the use of a tightly constraining morph trigram model. At the same time, the grammar size will be significantly reduced, benefiting the FST implementation.

5.4.2 The ANGIE FST

There are many ways in which the context-free formalism of ANGIE could be represented in FST form. We contend with the need to capture the ANGIE spacio-temporal probability space succinctly, maximizing both flexibility and compactness. Here, we describe an initial approach that was selected for its ease of implementation. In Section 3.7.1, we mentioned that a tree-like FST branching structure can capture all the alternative pronunciations of each vocabulary word. This FST which transduces phonetic sequences to their matching morphs, would enumerate multiple pronunciations associated with alternate ANGIE parses of each morph in a compact structure. All the pronunciation variants of each morph are gathered from the training data. The FST arc weights are compiled from ANGIE probabilities pre-computed during the training stage. The tree-like configuration allows for sharing of the FST arcs by common phonetic sequences, thereby reducing space required to represent all the allowable alternatives.

Recall from Section 3.7.1 that in this FST structure, two options are available to us: a left-to-right or a right-to-left branching network. In the left-to-right branching style, common phone sequences are shared advancing from the left, with the lexical symbols being emitted at the end of a sequence. The opposite occurs in the right-to-left branching style where sequences are shared from the right and lexical symbols are emitted at the beginning of the phone sequence. During the course of our investigations, we felt that it was more intuitively pleasing to choose a left-to-right configuration. As the recognizer proceeds left to right, it indeed seems more natural to assume a left-to-right branching framework where the lexical units are only known with certainty at the end of the phonetic sequence. However, we ultimately adopted the right-to-left style due to computational concerns. During FST composition of the ANGIE-FST with the morph n-gram model, the delayed output of the left-to-right network caused the creation of many superfluous dead-end states which, due to the specifics of the algorithm, could only be cleaned up at the termination of composition. Thus, this over-generation of extraneous states excessively consumed memory resources, causing the FST composition to fail. This did not occur in the right-to-left branching alternative.

Constructing the FST involves an initial phase of training the ANGIE grammar and compiling ANGIE probabilities in the usual manner. Then, all possible phone transitions for all morphs are discovered from the training data, and collapsed into a right-to-left branching tree structure. Figure 5-3 depicts an example FST for the *bos+* morph whose parse tree was illustrated in Figure 5-2. As analogous with a standard pronunciation graph, the structure will map out all the allowable variations on pronunciation for each lexical morph during recognition time. Which transitions can take place in the FST is entirely mandated by the existence of training instances, and the scores will be dictated by the trained ANGIE grammar. ANGIE phone terminals are utilized as the input alphabet, and the output strings comprise the morph lexicon. In the left-to-right branching structure, common phone sequences from the right end are collapsed onto common arcs. And morph labels are emitted at the beginning (on the left-most end of a branch). A self-looping arc

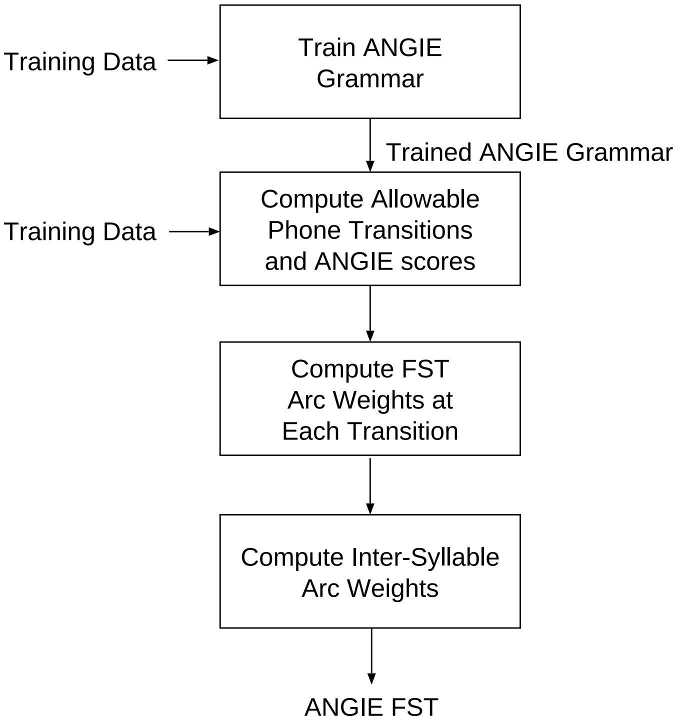
Training Data
Train ANGIE Grammar
Trained ANGIE Grammar
Training Data
Compute Allowable Phone Transitions and ANGIE scores
Compute FST Arc Weights at Each Transition
Compute Inter-Syllable Arc Weights
ANGIE FST

Figure 5-4: *Step by Step Summary of FST Generation Procedure.*

brings the path back to the start of the next morph to begin the next phonetic sequence. A guide to the FST generation algorithm is provided in the next section.

5.4.3 FST Generation Procedure

To begin with, a set of automatically generated forced alignments is required both to attain a trained ANGIE grammar and as training data for the FST generation. As done previously, this is obtained by seeding on a baseline SUMMIT recognizer. The step-by-step procedure (illustrated in Figure 5-4) used to compute the scores and phonetic transitions is set out below.

1. **Train grammar.**

   An ANGIE grammar is trained from a set of training utterances.

2. **Compute allowable phone transitions.**

   A second pass through this training set is performed to compile a structure that accounts for all phonetic sequences that occurred within training. For each morph, an ANGIE parse tree is obtained for the respective phonetic realization. The columns of the parse tree are recorded from the rightmost end. A left-branching structure is successively compiled by collapsing columns of the same phone terminals, advancing backwards (from right to the left).

At the same time, the score for each partial phonetic sequence as we traverse from the right across the graph is recorded in an array. The score for any partial phonetic sequence is computed by summing the corresponding column scores. Identical partial phonetic sequences may differ in score because, where the corresponding morphs differ, the associated ANGIE parse trees are also different, and therefore the respective columns score[3] differently.

3. **Compute FST arc weights from ANGIE scores.**

   Successively compute the arc scores from right to left in a recursive algorithm:

   ```
   Initialization.
   Compute_scores_for_left_arcs_of(R):
      foreach (A = left_arc_of(R)):
         arc_score(A) = max_score(A) - arc_score(R)
         Compute_scores_for_left_arcs_of(A)
   ```

   where

   $$max\_score(A) = max\{s_j, j = 1, 2, ..\} \tag{5.1}$$

   and s_j is the jth score for a particular phonetic sequence that is found by traversing the FST backwards from the end phone up to arc A. Essentially, this algorithm computes the best incremental score, among all the different ANGIE partial parses, for choosing the arc transition as we traverse from right to left. At any arc within the FST, the score of a path to the end will be the best score among all parses recorded. As we complete a phonetic sequence of a particular morph, the total path score will sum towards the total morph score from an ANGIE parse tree.

4. **Compute inter-syllable arc transitions.**

[3] Definition for a column score is found in Chapter 3.

The transition from the end of a syllable to the first phone of the next contains an arc weight that is a simple phone bigram probability computed at syllable boundaries. This is identical to ANGIE's treatment of syllable boundary probabilities.

After the construction of the ANGIE-FST, it is composed with the language model FST. The subsequent FST is determinized and minimized. This resultant FST combines both the language model and ANGIE probabilities, and is uploaded by the recognizer at recognition time.

5.4.4 Additional Comments

As the recognizer traverses the ANGIE-FST, the search space is perturbed by the FST arc weights which encapsulate the hierarchical constraints of the original ANGIE. Along a single branch or path, the arc weights are redistributed among the arcs, although the total score will exactly replicate the highest ANGIE parse score found for this phonetic sequence. The resulting score will reflect the likelihood of that particular pronunciation. Note that the compactness has been achieved by discarding much information such as the nodes in the ANGIE hierarchy and multiple alternate parses. Most of all, only the observations realized in the training data are recorded, and sequences that have not been previously manifested will not be supported. In fact, we have essentially ignored the remainder of ANGIE's vast probability space, and foregone its powerful generalizing abilities. An immediate problem is that of data sparsity. As a short-term solution, we have ensured adequate coverage by generating additional phonetic sequences to artificially boost the training data. Using the trained grammar, ANGIE is operated in generation mode to output sequences for each morph in its lexicon. These sequences are licensed by the dynamic parser but may not have occurred in training. Hence, we expect this to aid robustness to our FST. We shall return to the discussion on the FST structure and its shortcomings later in this chapter.

5.5 Stage Two

The second-stage system is very similar to the one previously described in Chapter 4. Again the algorithm can be run in two modes (1) the ANGIE system only and (2) the ANGIE-TINA integrated search. For the search routine, the same stack decoder is used to coordinate the partial theories proposed by both of the TINA and ANGIE parsers. As described in Section 4.6.3, the search directly implements a robust parse mechanism for TINA.

The fundamental difference from the previous system is the nature of the acoustic-phonetic network input. These networks are now in FST format and devoid of timing information[4]. The FSTs are topologically sorted such that each node corresponds with one time boundary, and arcs beginning/ending at the same node correspond with segments that begin/end at the same boundary. This does not impact upon the stack search which simply proceeds from left to right along these sorted nodes, though, loss of timing information forces us to abandon the use of duration models. On the other hand, by applying the FST optimization algorithms prior to the search, the network sizes are optimized, contributing to the search efficiency.

Let us highlight the difference in the application of ANGIE in the first versus second stage. In the second stage, a full word-level ANGIE grammar is used which therefore provides additional information from the morphological layer, not supplied in the earlier stage. Secondly, the dynamic ANGIE parser can generate a much richer probability space than the ANGIE-FST employed in stage one. Given a small phonetic network as input, a more complete set of hypotheses is supported by the ANGIE parser than by an ANGIE-FST. Hence, it is in our interest not to convert the second stage into an FST paradigm but to allow the full dynamic parse mechanisms of both ANGIE and TINA to interact and perturb the search space.

5.6 Experiments

5.6.1 Method

As we did in Chapter 4, a set of recognition experiments in JUPITER is conducted. The training corpus, the word lexicon and the second stage ANGIE and TINA grammars are identical to the ones used earlier. For the first stage, the morph-level grammar is much smaller than the word-level grammar. The number of unique ANGIE columns that are instantiated by the training data for the word grammar is 748, whereas this is only 413 in the morph-level grammar. A smaller number of columns indicates a smaller grammar

[4] If timing information were retained, it would decrease opportunities for merging identical theories, and therefore balloon the search space.

and a reduced number of parameters in our probability model. An ANGIE-FST is generated from the morph-level ANGIE grammar. It contains 7873 arcs and 1540 states. There are approximately 2.3 alternate pronunciations per morph. The phonetic networks from the first stage are restricted to a maximum of 1000 arcs.

These experiments will be compared with the single-stage SUMMIT baseline systems used in the previous chapter. To recapitulate, these are:

- **System 1 SUMMIT Top 1:** The best scoring sentence candidate.
- **System 2 SUMMIT N-Best ($N = 10$):** The most likely sentence according to a TINA-based NL post-processor.

As for the evaluation, we report on both understanding error rate (UER) and word error rate (WER). Experimental results under several conditions are ascertained for the same 352-sized set that was quoted in Chapter 4. For now, we refer to this set as a development set. Note that the parameters for this system were determined on another held-out development set. Finally, in order to ensure the validity of our results, the system is tested on a second previously unseen test set of 362 utterances.

For comparison purposes, we compose the ANGIE-FST with both a bigram (Systems 3 and 4 in Table 5-1) and a trigram (Systems 5 and 6). In both cases, the ANGIE-FST is pre-composed and optimized with the language model, leading to faster computation at run-time. The pre-composed FST with a bigram contained approximately 450,000 arcs and the one with a trigram contained approximately 2 million arcs. For the second stage, we compare the successive gains from augmenting with ANGIE only (Systems 3 and 5) and with ANGIE-TINA fully deployed (Systems 4 and 6).

5.6.2 Results and Analysis

Table 5-1 depicts results for the development set. In our previous work in Chapter 4, it was shown that a second ANGIE-TINA stage can enhance understanding performance on a word-level first-pass recognizer as well as recover performance losses incurred when the first stage was stripped of word-level constraints. The current results also reflect the same trends. System 3, using ANGIE only and a morph bigram, already performs comparably with the baselines, Systems 1 and 2. This is in spite of the relatively weak constraints of the morph bigram model of the first pass. In particular, System 3 equals the baseline System 1 in

System	WER (%)	UER (%)
1. SUMMIT Top 1	12.3	19.4
2. SUMMIT 10 Best	13.4	17.0
3. ANGIE (Bigram)	12.3	17.0
4. ANGIE-TINA (Bigram)	12.6	14.5
5. ANGIE (Trigram)	10.4	13.4
6. ANGIE-TINA (Trigram)	10.7	12.2

Table 5-1: *Word and Understanding Error Rates for the Development Set.*

System	MER (%)
1. SUMMIT Top 1	10.8
5 & 6. Trigram First Stage Top 1	9.7
5. ANGIE only	9.3

Table 5-2: *Morph Error Rates for the Development Set. The first line, the SUMMIT top 1, is the top scoring sentence from the single-stage baseline. The second line gives the morph output of the first-stage recognizer that is used in the two-stage architecture of Systems 5 and 6. A morph trigram is used in the first stage. The third line is the second stage output of the two-stage system of System 5, where ANGIE is used without TINA.*

WER but improves on UER. This would suggest that the ANGIE-enhanced first stage has an even greater impact on understanding than word accuracy. When TINA is added in the second stage, UER drops significantly. This offers a 14.7% relative reduction to UER (from 17.0% to 14.5%) compared with System 2.

When a morph trigram is used in the first stage, even more improvement is achieved with both ANGIE only (System 5) and ANGIE-TINA (System 6). Again examine the Systems 1 and 5 which do not employ NL. System 5 gains 15.4% relative reduction in WER (from 12.3% to 10.4%) and, much more substantially, a 30.9% relative reduction in UER (from 19.4% to 13.4%). This reinforces the observation that the ANGIE sublexical models have greatly benefited understanding. The best understanding performance is achieved by System 6, with a 28.5% relative reduction to UER (from 17.0% to 12.2%), compared with System 2. Systems 5 and 6 outperform all the results quoted in the previous chapter. Also, to further validate our observations in Chapter 4, adding TINA in the second stage consistently decreases UER, accompanied by a small WER degradation.

These encouraging performance gains can largely be attributed to the ANGIE scores incorporated into stage one. This is further substantiated when we consider the morph

System	WER (%)	UER (%)
2. SUMMIT 10 Best	14.9	20.8
6. ANGIE-TINA (Trigram)	11.2	15.9

Table 5-3: *Word and Understanding Error Rates for the Test Set.*

error rates (MER) in Table 5-2. MER is computed as defined in Chapter 4. We report the MER for the baseline System 1 and for the best scoring hypothesis in the first pass when a morph trigram is used. There is a 10.2% improvement (from 10.8% to 9.7%.) This suggests that considering the first stage alone, a morph-based ANGIE-FST recognizer is superior to a word-based baseline recognizer, a reflection on the power of ANGIE's sublexical constraints when combined with a morph trigram. When we consider the MER of the second stage output (in ANGIE only) at 9.3%, we can directly infer that the word-level ANGIE grammar still contributes additional knowledge, and enhances performance through its use of higher-level information, even though much performance improvement was already reaped in the first stage by the morph-based ANGIE.

The acoustic scores on the ANGIE-FST arcs are weighted by a scaling constant which is determined on a separate held-out development set. During the investigation, we noticed that an excessively large scaling factor causes the recognizer to proliferate hypotheses of commonly occurring subword structures, such as morphs pertaining to mono-syllabic function words. By contrast, an overly small scaling factor translates to a significant performance degradation. Although the bigram and acoustic models supply the constraint, multiple alternate pronunciations presented by the FST confound the recognizer search. Often wrongful hypotheses associated with unlikely pronunciations are selected. We conclude that the ANGIE probabilities are crucial in steering the search through many possibilities of pronunciation variation. This is most beneficial for recognizing spontaneous speech.

To further validitate our findings, we attained more results on an independent test set of 362 utterances. The results are set out in Table 5-3. When comparing with the 10-best baseline, a 23.6% relative reduction (from 20.8% to 15.95%) in UER was achieved using a morph trigram ANGIE-FST first pass with ANGIE-TINA second pass.

5.7 Discussion

The above experimental results have established a promising start to a powerful low-level recognition engine. Most of our gains are centered on augmenting the first stage with ANGIE. In part, we have overcome the hurdle of incorporating a highly complex set of hierarchical models efficiently into the recognizer. And this was accomplished within near-real-time. As a result, the first stage is endowed with a highly effective probabilistic mechanism for modeling pronunciation variability embedded in a much more tractable FST structure. In summary, this system has exhibited performance gains over the baseline for the following reasons:

- Subword patterns including phonological processes have been automatically learned from the training data and modeled statistically.
- Then, compared with the baseline, a larger number of alternative pronunciations are allowed. The probability models effectively guide the recognizer to select the correct one.
- Furthermore, the ANGIE-FST implicitly encompasses constraint from the longer distance information that ANGIE embodies.

The above points hold even when using the smaller syllable-level ANGIE grammar. This demonstrates that much relevant contextual information resides at the syllable phonotactics and phonological levels. To our satisfaction, the results indicate that benefits gleaned from ANGIE are reflected more so in terms of understanding performance. In our view, understanding improvement is critically more important than word accuracy.

However, let us proceed to identify a key remaining issue: the FST generation relies on memorizing observations in precise sequences from training data. This has gained us compactness in representation by only recording a portion of the ANGIE probability space. Nonetheless, it is fundamentally at variance with the philosophy of ANGIE's design, which is to predict the likelihood of sequences that are *not* instantiated during training using *implicit* knowledge gathered from training. The FST does not fulfill this as it fails to accept phonetic sequences that have not been previously encountered. It *explicitly* captures only patterns from the training set. Instead of generating probabilities on-the-fly like ANGIE, probabilities are pre-computed and assigned to the FST arcs. A temporary solution to address this has been to utilize additional training data, attained from using ANGIE in generation mode. The artificial training data simulate realizations that capture a larger piece of the ANGIE probability space, thereby enhancing FST coverage. Although this is an *ad hoc* method, it suffices in the meantime for alleviating sparse data problems associated with alternative pronunciations of in-vocabulary data. And this has been reflected by the successful results in our experiments. But nevertheless, it inherently lacks the dynamic and generative characteristics underlying ANGIE.

On closer examination, some fundamental limitations revolve around the right-to-left branching configuration. Observe that all the vocabulary items (in our case, morphs) are emitted at the beginning in time. This was designed in the interest of optimizing the composed FST size. During composition with the n-gram model FST, ANGIE and n-gram scores are combined at the beginning of the phonetic sequences. In this way, early in the search path, the recognizer is forced to propose in-vocabulary word candidates, and is precluded from considering an unknown word possibility. It translates to a top-down approach, as opposed to the bottom-up method where ANGIE is called to propose likely word candidates, at the end of phonetic sequences.

Moreover, the structure embodies none of the sharing of common subword structures performed within ANGIE. Quite the opposite, it "undoes" this, as all the morphs, including the numerous homomorphs, are represented by distinct branches in the FST. The size of the FST grows in proportion with both the size of the morph vocabulary and the training data. Such redundancy seems to contribute to inefficiency.

Ultimately, this FST is inextricably confined to the pre-determined recognizer vocabulary which in itself is a contextually rich set of morph units, with too much reliance on the topic domain. In order to support OOV words, more flexibility would inevitably be necessary. This should necessarily be achieved by both introducing more generality in the lexical units, and rethinking the way the FST paradigm portrays ANGIE.

5.8 Final Remarks

In the next chapters, we pursue the path of placing more domain-independence in the front-end towards the goal of adding more flexibility and generality. The first stage will need to support novel word constructs, that are independent of any fixed word lexicon. Chapter 6 will focus on a new enhanced ANGIE grammar and a re-implementation of the ANGIE-FST.

These will enrich the contextual information of the first stage, while moving away from dependence on high-level domain-specific knowledge.

Chapter 6

Improvements for the First Stage

6.1 Overview

This chapter will examine in depth the development of two innovative ideas for the first stage. These are (1) the use of ANGIE to simultaneously model grapheme information in conjunction with phonological and other sublexical knowledge, and (2) a new method for encapsulating ANGIE in an FST which can support previously unseen sequences. In Section 6.2, the principal motivating issues underlying these ideas are highlighted. In Section 6.3, our methodology for incorporating spelling into the ANGIE grammar is detailed. This is accompanied by an explanation of the *letter-phoneme* units, invented for this grammar. The following section will proceed to explicate the new FST structure, the *column bigram*. Here, we outline the algorithm used to generate this FST, and explore the implications of this new flexible design. Section 6.5 considers a revised two-stage system with the new features incorporated. We present some preliminary results from a JUPITER recognition experiment using only in-vocabulary data.

6.2 Design Considerations

Let us identify two remaining design issues:

1. Maximizing linguistic constraint: introducing more low-level language constraints to improve performance.

2. Increasing flexibility and generality: allowing the recognition of phonetic sequences

that have not occurred during training as they pertain to OOV words.

These are dual goals that critically impact our system's capability to handle unknown words or dynamic vocabulary. Yet they are conflicting. Tightening constraints, such as by using more contextual knowledge, usually hampers flexibility by heightening domain specificity. (This would oppose our goal for a domain-independent first stage.) Alternatively, allowing more flexibility in the linguistic models, such as through using more general units (for instance syllables), usually involves a relaxation of constraints. In doing so, we forego some contextual information, leading to deterioration in performance on in-vocabulary data. For the systems presented in previous chapters, excellent performances on in-vocabulary sentences have been observed. But these designs would poorly handle the incidence of unknown words, let alone dynamic vocabulary. Hence, the focus needs to be on adding more flexibility without losing these performance gains.

Our first measure is to augment the linguistic constraints with more low-level domain-independent knowledge. One such source of knowledge can be found in grapheme information. In Chapter 1, we mentioned that grapheme information has rarely been used in the past to build acoustic models directly. On these occasions, the graphemic symbol set was used in place of the phoneme set. Naturally, this led to an unacceptable drop in performance. In our case, we argue for using a set of linguistic units that embody both graphemic and phonemic properties. By loading our mode of representation with richer contextual knowledge, we attain finer-grained models which remain, to a large extent, domain-independent. These units are integrated within the ANGIE framework in such a way that the pre-terminal to terminal layer serves to characterize phonological processes as before but also models sound-to-letter rules. The result is that the grapheme information is combined seamlessly with sublexical information to predict phonetic sequences.

In Chapter 5, we demonstrated the success of using an ANGIE-FST, but it was clear that a new FST structure is needed for our purposes. In particular, we desire an FST structure which firstly covers the space of possible parses that is spanned by ANGIE, and secondly allows some access to the elements of the internal parse structure. The first point is critical in accepting previously unseen sequences from unknown words that would be supported by the ANGIE parse mechanism. As will be seen, this is accomplished by adopting an ANGIE column bigram method.

A third point of consideration is the set of lexical units in the first stage. Thus far, we have relied upon a set of highly context-rich morph units. As these potentially combine to form many novel words not set out in the existing JUPITER word lexicon, they offer some generality. But in actuality, they are inadequate in supporting sequences from the vast possibility of unknown words. We will see in Section 6.5 that a new set of *sub-morph* units are derived by splitting stressed root morphemes into smaller component units. This will preserve much of the predictive power afforded originally by this lexicon but also expands its power to generalize where it is most necessary.

Finally, it must be noted that the first stage outputs a phonetic network. That is, the morph hypotheses of the first stage are reduced to the corresponding phonetic sequences. This can be viewed as particularly beneficial as the system is not committed to the early hypotheses derived solely on low-level linguistic knowledge. Although these knowledge sources need to be reliable to select the correct portion of the search space, the contextual information, including the morph sequences, the spelling hypotheses and the ANGIE parses, is abandoned. The function of the first stage is simply to partially prune the search space, whereas the second stage has the opportunity to recover any errors by returning to the phones, the most basic atomic units of the recognizer, and using more powerful, word-level information from the outset.

6.3 A Spelling-Based ANGIE Grammar

6.3.1 Introduction

There are two potential benefits for incorporating grapheme information. First, by equipping models with spelling knowledge, we could directly deduce spellings from phonetic hypotheses at unknown words. Spellings can be accessed within the recognizer models, thus obviating the need for a separate sound-to-letter module. We imagine that having quick access to putative spellings for an unknown word can provide intuitive guesses for its possible identity. Secondly, as already mentioned, we may exploit spelling information as another form of low-level domain-independent linguistic constraint.

Given ANGIE's rich probabilistic structure and its potential to be integrated seamlessly within our recognizer, we intend to use it as a mechanism to encode the grapheme information. This will allow graphemics to combine and interact with other sublexical phenomena to predict phonetic sequences. In the past, ANGIE has already been applied to the dual tasks of sound-to-letter/letter-to-sound conversion, in which letter units are used in lieu of phones at the terminal layer of the parse tree. The success gained there has reaffirmed that orthography can systematically exhibit cues that reflect structural information such as morphology, syllabification and phonemics. And this relationship between sublexical patterns and graphemics can be captured and exploited in the ANGIE parse tree. Moreover, this was achieved in spite of the extent of irregularities in English, which contains many variations from dialects, borrowings from other languages and so forth.

We surmise that graphemics can aid prediction of phonetic sequences when *combined* with the multi-layer sublexical models in ANGIE. In fact, our strategy is to incorporate grapheme information in the context-free grammar of ANGIE, effectively integrating the knowledge within the pre-terminal units. This differs from past uses of graphemics [2, 92] as it does not replace phoneme-based models but enhances the existing linguistic framework, preserving the terminal layer for incoming phonetic sequences. The resulting parse tree characterizes generic word substructures, phonological processes and sound-to-letter conversion within the same probability models. The units of the pre-terminal layer now form a new set of symbols called *letter-phonemes* that embody both pronunciation and spelling, in place of the more conventional phonemic set that was previously used. The next section will comprehensively detail the development of the letter-phonemes.

6.3.2 Letter-Phonemes

In the ANGIE grammar, the phoneme set, that resides at the pre-terminal layer, has been expanded and molded into a new symbolic representation, the *letter-phoneme* set. These letter-phonemes are designed by annotating letter units with carefully chosen characteristics that distinguish phonemic correspondence and other linguistic properties, including stress, context and syllable position. To familiarize the reader, we provide two initial examples below.

Stressed Long Vowels	a_l+, ai_l+, ay_l+, e_l+, ea_l+, eaux_l+, ee_l+, ei_l+, eigh_l+, eille_l+, ew_l+, ey_l+, i_l+, iew_l+, igh_l+, ioux_l+, is_l+, o_l+, oa_l+, oe_l+, oh_l+, oo_l+, ou_l+, ow_l+, u_l+, ue_l+, ul_l+, y_l+, ye_l+, yu_l+
Stressed Lax Vowels	a_x+, ai_x+, e_x+, ea_x+, i_x+, o_x+, oo_x+, or_x+, ou_x+, u_x+, y_x+
Stressed /l/ and /r/ Colored Vowels	aal+, ahr+, air+, aire+, al+, all+, ar+, are+, arr+, aul+, aw+, e+re+, ear+, eir+, el+, elh+, ell+, elle+, eoul+, er+, ere+, eur+, il+, ill+, ille+, ir+, ol+, ole+, oll+, oor+, or+, ore+, orr+, oul+, our+, owl+, r+, re+, uer+, ur+, urr+, yr+
Other Stressed Vowels	a+, ah+, ao+, as+, au+, aw+, eh+, o+, oi+, ois+, ou+, ow+
Unstressed Rhymes	a_uns, ah_uns, ai_uns, air_uns, al_uns, an_uns, ar_uns, au_uns, ay_uns, e_uns, ea_uns, eau_uns, ee_uns, eigh_uns, el_uns, ell_uns, en_uns, er_uns, es_uns, eu_uns, ew_uns, ey_uns, i_uns, ia_uns, ie_uns, il_uns, ille_uns, in_uns, ing_uns, ir_uns, ire_uns, le_uns, o_uns, oe_uns, ol_uns, on_uns, oo_uns, or_uns, ou_uns, our_uns, ow_uns, r_uns,re_uns, u_uns, ul_uns, ur_uns, ure_uns, y_uns, yl_uns, yu_uns
Onset Consonants	b!, c!, ce!, ch!, cs!, d!, er!, f!, g!, ge!, gi!, h!, j!, ju!, k!, kn!, l!, lh!, m!, n!, o!, p!, ph!, pp!, qu!, r!, rh!, s!, sh!, ss!, su!, t!, tch!, th!, ti!, tw!, u!, v!, w!, wh!, y!, z!, zh!
Non-onset Consonants	+ve, b, bb, be, c, ce, ces, ch, ck, ct, d, d*ed, de, dge, dne, f, fe, ff, g, ge, gg, gh, gi, is, k, ke, l, le, ll, m, me, mm, n, n+t, nch, nd, ne, ng, nk, nn, nt, p, pe, ph, pp, r, s, s*pl, se, sh, she, sk, ss, st, t, tch, te, th, the, ti, tt, tte, tts, u, v, ve, w, x, y, z, ze
Function Words	a_ey, a_fcn, ai_fcn, are_fcn, e_fcn, e_the, eah_fcn, ee_fcn, en_fcn, ere_fcn, ey_fcn, i_i, i_fcn, ia_fcn, ill_fcn, o_fcn, o_to, oe_fcn, or_fcn, oul_fcn, our_fcn, ro_fcn, u_fcn, uh_fcn, y_fcn, you_fcn

Table 6-1: *A List of Letter-Phoneme Categories. These are arranged in the major phonemic categories. Stressed vowels are appended with a "+". "_l" appends a long vowel. "_x" appends a lax vowel. "_uns" appends an unstressed rhyme. "!" appends a consonant in onset position. "_fcn" appends vowels in function words. See text for further explanation and see Appendix B for a list of meanings of annotations.*

The letter-phoneme */ti!/*[1] represents a consonant in the onset position (annotated by an *"!"*) which is spelled with the letter sequences *ti*. This is exemplified by words such as *condition*, *precipitation* and *question*. Hence, the phonemic representation can be the palatal fricative */sh/* or the affricate */ch/*. As with the original grammar, a set of hand-written context-free rules specifies the allowable phonetic realizations of each letter-phoneme. The rule associated with */ti!/* is given by[2]:

$$\begin{array}{lcl} \textit{ti!} & \Rightarrow & \text{[jh] (sh ch)} \\ & & \text{[tcl] ch} \end{array}$$

This rule restricts the phonetic realization to a /sh/ or /ch/ phone, where each can be optionally preceded by a /jh/ phone. The /ch/ phone may be preceded by the stop closure /tcl/.

The letter-phoneme is also associated with a high-level rule:

$$\text{UONSET} \Rightarrow (\ldots\ \textit{ti!}\ \ldots)$$

which specifies that this unit can be a child node of the category UONSET, for an onset consonant in an *unstressed* syllable.

[1] Letter-phonemes are depicted much like phonemes in our notation. They are italicized and enclosed in "//"s.

[2] In our notation for a context-free rule, alternative symbols are enclosed in parentheses ("()"s), and optional symbols are enclosed in brackets ("[]"s). Enclosing "//"s are omitted on the units on the left and right hand sides of the rule. This is done for clarity.

Another example is the letter-phoneme */ear+/*. Phonemically, this unit covers two adjacent units, representing a vowel followed by an */r/* phoneme. The *"+"* indicates that this letter-phoneme appears in a lexically stressed syllable. The context-free rule associated with */ear+/* is shown below by:

$$\begin{array}{lcl} \textit{ear+} & \Rightarrow & \text{ihr (rx r)} \\ & & \text{ehr (rx r)} \\ & & \text{er [r]} \end{array}$$

On the right-hand side are the phonetic units that correspond with those in the acoustic models used in the recognizer. */ihr/* is a context-dependent */ih/* phone followed by retroflexion. Similarly, */ehr/* is a context-dependent */eh/* phone followed by retroflexion. The first line of the rule is applicable when */ear+/* appears in the word *pearson*. The second line is applicable in the word *wear*. And the last line is applicable for */ear+/* in the word *heard*. */ear+/* is also associated with the following high-level rule that specifies that it can only appear within a stressed nucleus:

$$\text{NUC+} \Rightarrow (\ldots\ \textit{/ear+/}\ \ldots)$$

Effectively, these letter-phonemes are subdividing the phoneme space into more specific units, resulting in finer-grained probability modeling. Previously the phoneme-to-phone layers, by and large, capture phonological processes. But now the functionality of the preterminal to terminal layers has widened to capture sound-to-letter rules as well, thereby exerting tighter constraint.

As the lexicon in ANGIE is organized into two tiers, vocabulary words are defined in terms of their morph baseforms, whereas morphs are defined by their phonemic sequences. For the new grammar, every morph is associated with a letter-phoneme sequence in the baseform. In some cases, a morph can be associated with more than one letter-phoneme sequence to represent multiple alternate pronunciations. Table 6-2 exemplifies a morph lexicon for the stressed roots *rain+*, *cane+* and *can+*, and a suffix *-tion*. From the lexicon, both

Morph Lexicon		
can+	:	*c! a+ n*
cane+	:	*c! a_l+ ne*
rain+	:	*r! ai_l+ n*
-tion	:	*ti! on_uns*

Table 6-2: *An Excerpt from the Letter-Phoneme Morph Lexicon. Example morphs are the stressed roots can+, cane+ and rain+ and the derivational suffix, -tion.*

the morph spelling and pronunciation can be directly inferred. As observed in the table, letter-phoneme sequences can be stripped of their annotations and concatenated together to synthesize the orthography. Meanwhile, the annotations encode phonemic associations[3]. This attribute would be particularly useful during recognition time. It is envisioned that a system may encounter a novel letter-phoneme sequence from an unknown word, and a potential spelling is instantaneously deduced.

All the categories for the 289 letter-phonemes are listed in Table 6-1. These categories are chosen by hand in an attempt to reduce perplexity and improve predictive performance in both letter-to-sound and pronunciation variation. More examples of their associated context-free rules are provided in Appendix C. Following are some properties that these units describe:

- **Vowels:**
  - Lexically stressed vowels are denoted by a suffix marker, *"+"*.
  - Long stressed vowels are appended with a *"_l+"*. For example, */i_l+/* is the letter *i* which is phonetically realized as a long stressed vowel, that is, it maps to either */ay/* or */iy/* in a stressed syllable.
  - Lax and stressed vowels are denoted by a *"_x+"*. For example, */ea_x+/* is the grapheme *ea* which is realized as a lax stressed vowel, that is, it maps to */eh/*, in a stressed syllable, as in the word *head.*
  - Some units correspond with syllable nuclei or rhymes that are not lexically stressed. These are annotated by an appending *"_uns"*.
  - A subset of letter-phonemes correspond with pseudo-diphthongs. These include

[3] The next subsection will elaborate on the range of properties that the letter-phonemes covered in their diacritics.

  /r/ and /l/-colored vowels. Some examples are */oul+/* and */owl+/*.

  - As with the original phoneme grammar, some vowels within function words are modeled separately. For example, */ee_fcn/* appears as the vowel within the function word *been*. Some vowels are specific to the word-context. Examples are */o_to/* for the vowel in *to* and */i_i/* for the vowel in the word *I*.

- **Consonants:**

  - Consonants that occur in the syllable-onset position are marked with a *"!"*.
  - Some letter-phonemes capture diphone contexts within a single unit. Some examples are */nch, nd, nk, ck, ch/*.
  - As in the phoneme grammar, some units are specific to the associated inflectional suffix. These are */d*ed/* for a past tense, spelled with a */d/* or */ed/* and */s*pl/* for the plural suffix.

- It is worth noting that some letter-phonemes have distinctly differing phonemic correlates such as the coda */gh/* which can be realized phonemically as */f/* or */g/*. We will see further on that the correct phonemic realization is learned through the probability models.
- Multiple letter-phonemes may have the same phonetic realization. For example, */n/* and */ne/* are different spellings for */n/* in the coda position. Similarly */s/* and */ss/* also translate to the same phonemic pronunciations.

It is apparent that the letter-phoneme representation is much more expansive, accommodating many more contextual properties when compared with a phoneme set. This also invites some sparse data problems. In the instances where training incidences are sparse, some graphemes are collapsed together into a single letter-phoneme, forming a more generic model. For example, */ain/* and */oln/* (that appear in the words *mountain* and *lincoln*, respectively,) are spelling variations of an unstressed rhyme realized as */en/*. They are merged together into one model due to insufficient data. In doing this, some spelling information is discarded and cannot be recovered. This amounts to a compromise on the part of the sound-to-letter predictive power.

It must be highlighted that the ANGIE framework can uniquely exploit the enhanced features of this letter-phoneme set. We have especially fine-tuned by hand our selection of these units so that the probability models can optimally predict both pronunciation and spelling during parsing. Let us illustrate this with some examples. Referring to Table 6-2, consider the baseform sequences for the morphs *can+* and *cane+*. According to the context-free rules */a+/* can be realized by an /ae/ or /aa/, and */a_l+/* can be realized as an /ey/ only. The rules are specified as follows:

$$a\_l+ \;\Rightarrow\; \text{ey}$$

$$a+ \;\Rightarrow\; (\text{ae aa})$$

Also the letter-phonemes reside at the right-hand side of the following high-level rules:

$$\text{LNUC+} \;\Rightarrow\; (\ldots\; a\_l+\; \ldots)$$

$$\text{NUC+} \;\Rightarrow\; (\ldots\; a+\; \ldots)$$

The last two columns of the parse trees that correspond with the rhyme in *can+* and *cane+* are depicted adjacent to each other below.

STRESSED ROOT		STRESSED ROOT	
NUC+	CODA	LNUC+	CODA
a+	*n*	*a_l+*	*ne*
ae	n	ey	n

As explained in Chapter 3, the parsing proceeds one column at a time and examines trigrams from the bottom upwards. The probability of a letter-phoneme is given by a letter-phoneme from the left column and the child phone. Comparing parses in *can+* and *cane+* above, observe that ANGIE will automatically learn from training data through probabilities $P(/n/|/a+/, /\text{n}/)$ $P(/ne/|/a\_l+/, /\text{n}/)$, that a long vowel will predict the next consonant n to be followed by an e. Alternatively speaking, the presence of the e ending in the spelling is an indicator for the vowel pronuncation. This is not specified in the context-free rules but is learned via the training procedure. Similarly in *rain*, the vowel has the associated rule given by

$$ai\_l+ \;\Rightarrow\; (\text{ey ay})$$

and the rhyme portion of the parse tree is given below. ANGIE learns from training that

STRESSED ROOT	
LNUC+	CODA
ai_l+	*n*
ey	n

in the context of a long vowel spelled with */ai/*, the letter *n* in the coda is not likely to be followed by an *e*.

6.4 The Column-Bigram FST

6.4.1 Introduction

In the previous chapter, the ANGIE-FST has demonstrated the utility of incorporating ANGIE knowledge in the first stage. But our remaining concern is the reliance on pre-computing ANGIE scores and memorizing training observations. Within this FST, allowable paths are limited to entire phonetic sequences that have been instantiated, precluding previously unseen sequences or rare pronunciation variants.

The new column bigram FST method is designed to align more closely with the underlying ANGIE philosophy. The desired FST must accept previously unobserved sequences in the same manner that ANGIE can parse OOV words. Furthermore, unseen or rare combinations must be supported as in ANGIE, which generalizes from the well-trained probabilities of word substructures observed at training time. With this in mind, the FST configuration should reflect the parse structure, and distribute ANGIE probabilities along the arcs accordingly. Equally important is the ability to access the ANGIE parse structure via the output symbols of the FST. A flexible algorithm can generate additional contextual information which may be useful in enforcing more long-distance constraints.

In the previous chapter, the right-to-left branching structure computed an FST represented by $P \circ L$ in the overall equation for the recognition task. Described below is the column bigram which will only compute the transducer P. This conducts a mapping from the phonetic sequences, hypothesized by the acoustics-driven search. We will see that the output alphabet of P can be determined with some flexibility. And P is composed with L which sets out the lexical units of the recognizer. Effectively, we distinguish between the transducer which performs the sublexical modeling P and one that specifies the lexical units L, whose determination is also somewhat flexible.

6.4.2 Column Bigram Structure

With the column bigram method, the ANGIE parse tree must be viewed as a sequence of columns, with phones at the terminal. Proceeding from one phone to the next in time corresponds with transitioning across columns from left to right. Effectively, the FST expresses bigram statistics on transitions of the columns of the ANGIE parse. In order to do this, it is necessary to compute the probability of generating one column given the previous left column context, while keeping in mind that we wish to replicate the probabilities in the original ANGIE probability models. Recall that ANGIE parse probabilities are given by (1) trigram bottom-up probabilities where the probability of a node is given by its left sibling context and child node context, and (2) advancement probabilities that produce probabilities for a phone terminal given the entire left column context. Recall also that the probability of a parse tree is composed of (log) summations of the abovementioned statistics. Consider that the probability of generating a column C_i given the previous C_{i-1} can be given in the following:

$$P(C_i|C_{i-1}) = P(n \in C_i|m \in C_{i-1}) \tag{6.1}$$

where n and m are nodes in columns C_i and C_{i-1} respectively. Then, given the independence assumptions taken that a terminal node is dependent on the previous column and the non-terminal nodes are only dependent on the left node contexts and child node contexts:

$$P(C_i|C_{i-1}) = \prod_{n \in C_i} P(n|n_l, n_c) \times P(p|C_{i-1}) \tag{6.2}$$

where n is a non-terminal node, n_l, situated within column C_{i-1}, is the left node context of n, n_c is a child node context of n and p is the terminal phone node of the C_i. Alternatively, in terms of log probabilities:

$$LogP(C_i|C_{i-1}) = \sum_{n \in C_i} LogP(n|n_l, n_c) + LogP(p|C_{i-1}) \tag{6.3}$$

The first term is the logarithmic sum of the trigram probabilities in the column and the second translates to the advancement probability.

Essentially, Equation 6.3 is a formulation for bigram probabilities on adjacent column pairs. As bigram models are commonly represented in FSTs, it is easy to construct an FST for column bigrams, upon computing the requisite column transistion probabilities. Let us describe the steps involved during the training phase.

1. **Train Grammar:**

   The first iteration through the training data trains up the ANGIE grammar and computes the ANGIE parse probabilities. Note that in the previous chapter, we dealt with a syllable-level grammar where information at the morphological layer was largely stripped away, and ANGIE was trained from a single lexicon of morph units. These morphs were precisely matched with those of the first-stage recognizer lexicon. In our new method, we are able to train up the original word-level grammar from word orthographies and phonetic alignments. And this ANGIE grammar uses the two-tiered lexicon structure introduced in Chapter 3. We will see soon that the units of the recognizer are related with the ANGIE lexicons but not confined to be the actual top tier words, defined and used by ANGIE in training.

2. **Compile ANGIE Columns and Transition Probabilities:**

   The second iteration reparses the training and enumerates all the unique ANGIE columns (with distinct nodes derived from the parse) that have been instantiated. For all adjacent column pairs that are observed, column bigram probabilities are computed and recorded.

3. **Construct Bigram FST:**

   Finally the bigram FST is constructed. This FST will consist of phonetic sequences defined for its input alphabet and pre-terminal sequences defined for its output alphabet. We selected the pre-terminal layer for output as this is the layer at which the lexical units are specified in ANGIE. These pre-terminal units can be either phonemes or letter-phonemes but, from here on, our experiments will involve using the letter-phoneme units exclusively. Moreover, at selected boundaries, additional information regarding the parse structure is emitted. We chose to output the morph class label at every morph boundary. At a stressed root, instead of emitting the morph class label, the component sub-syllabic labels, ONSET and RHYME[4] are emitted following the associated column. Outputting information extracted from the parse structure provides more linguistic context and exerts additional long-distance constraints. In fact, this information will disambiguate paths that correspond with distinct ANGIE parses, ensuring that the FST closely emulates ANGIE's models. A simplified schematic of the FST of one phone sequence is depicted below in Figure 6-1.

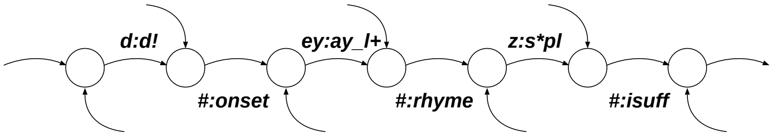

Figure 6-1: *Schematic of an FST for the Word Days. The input and output labels, given at the arcs, are delineated by a ":". The phonetic realization is given by /d ey z/, and the letter-phoneme baseform is given by /d! ay_l+ s*pl/. The "isuff" label is emitted indicating the inflectional suffix at the column associated with the plural and the /s*pl/ letter-phoneme.*

During the actual construction of the bigram FST, a new state is created for every unique ANGIE column. For a particular column, the outgoing arcs of that state represent transitions to other columns, and the respective bigram probabilities reside on the arc weights. Additional states are created to accommodate the emission of the morph class labels. Extending from every column that can terminate a word[5] are arcs that are connected with all columns which are allowed to begin words. On these arcs are simple word-boundary phone bigram probability estimates, as in the original ANGIE. Figure 6-2 illustrates a portion of an actual column bigram FST.

6.4.3 Coverage and Smoothing

The column bigram method is a significant departure from our previous FST generation method. Our foremost consideration in the re-design has been to obtain maximal coverage of the vast probability space of ANGIE, while preserving compactness in the FST representation. Let us assess our success in attaining this goal. The resultant FST is a flattened partial imprint of the complete grammar. All the distinct columns that have been instantiated by the training are recorded by the FST, and direct connections are drawn between the

[4] We define rhyme as the part of the stressed root that follows the consonant onset. Therefore all stressed roots contain a rhyme, even though the consonant onset may be absent. The rhyme contains a single vowel nucleus with an optional subsequent coda. See the glossary in Appendix A for a full list of definitions of linguistic terms.

[5] We are referring to a word as found in the top tier ANGIE word lexicon.

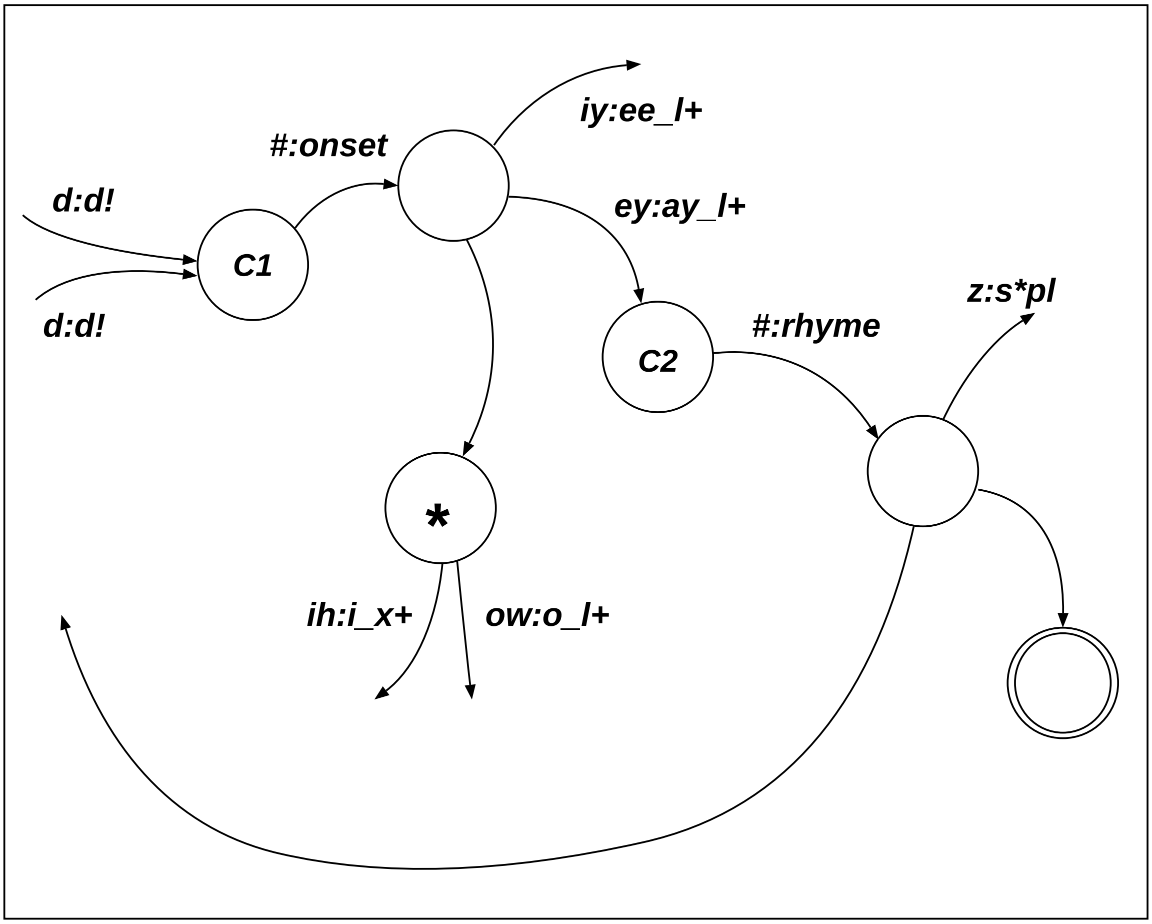

Figure 6-2: *Schematic of the Column Bigram Structure. A path for the word "days" has been captured in this FST. C_1 is the state that corresponds with a column with phone terminal /d/ and C_2 is the state that corresponds with a column with phone terminal /ey/. The state marked with "*" is the back-off state connecting from an onset column to other rhyme columns.*

states belonging to column pairs that have occurred in adjacency during training. Yet, this only constitutes a subspace of ANGIE's coverage, that is designated for the over-generalizing sublexical patterns. The reason is that, for one, the parse mechanism is capable of generating novel columns that are permitted by the context-free rules and are supported by the trigram probabilities. Our decision is to omit these for the time being, on the grounds that these columns would be unlikely and this would be reflected by very low scores. However, a second point of concern is the many transitions between column pairs that have not been manifested in training. This is a possible source of a sparse data problem, and calls for a solution to circumvent this. In the previous chapter, we implemented the temporary solution of using the parser to generate artifical training data in the hopes of covering more transitions. But here, a more robust strategy is to focus on addressing sparse transitions where it is most critical: at boundaries between morph units.

From preliminary investigations, many novel morph transitions are disallowed due to lack of training observations. It will become apparent that these transitions are particularly important because ultimately, our lexical units are morphs,[6] and it is crucial that novel sequences of morphs are permitted. To overcome any sparse data problems that may prevail at these boundaries, a simple back-off or smoothing mechanism has been implemented.

A back-off state corresponding with every morph class is created. This provides pathways for all the missing bigrams at morph boundaries. The schematic in Figure 6-2 includes a back-off state denoted by a "*". At any morph-final column C_i, the probability of transitioning to the back-off state S_b is computed as follows:

$$P(S_b|C_i) = 1 - \sum_j P(C_j|C_i) \tag{6.4}$$

where $P(C_j|C_i)$ represents the probability assigned to column C_j given its left context column C_i. In practice, the summation is computed by seeking the total probability of arcs exiting a column node, C_i, and assigning the remainder probability space to the transition towards the back-off node. This space corresponds with that which has been allocated towards unseen data by the ANGIE parse mechanism. As for exiting the back-off state, a maximum likelihood estimate for the probability of the next column given by left morph class context is computed. Although these smoothing probabilities are not part of the scheme in the original parser, they ensure that the important transitions are supported probabilistically. The back-off mechanism does not extend to function words nor does it connect between columns that are disallowed by the ANGIE rules. An example would be to directly transition from a prefix to a suffix, omitting a stressed root. This would constitute a parse failure in the original parser.

6.4.4 Discussion

The column bigram offers us many advantages over our previous approach to FST generation, because it shares with the original parser many important characteristics. The

[6] More precisely, we will see later that the lexical units are composed of unstressed morph units and stressed onsets and rhymes.

WORD			WORD			
STRESSED ROOT		ISUFF	STRESSED ROOT			
ONSET	NUC+	PLURAL	ONSET		NUC+	CODA
d!	ey+	s*pl	p!	l	ey+	s
d	ey	z	p	l	ey	s

Figure 6-3: *Tabular Schematic of ANGIE Parse Trees. The words represented are days and place.*

bigram structure resembles that of a phone bigram with the distinguishing property that the phones are embedded with entire column contexts that encompass much valuable long distance information. Some of this is emitted at the morph boundaries, and directly exerts constraint. Unlike our previous right-to-left branching FST, a full word grammar can be used for the training. That is, implicit word-level information trains up the sublexical models, even though at the lexical level, morphs will be used. (We explain the lexicon in the next section.)

Most importantly, the FST is trained from in-vocabulary data but extends that learned knowledge in order to apply to sequences that are previously unobserved. This is accomplished by memorizing column pairs rather than entire phone sequences. For example, in Figure 6-3, the parses for *days* and *place* are depicted. Earlier in Figure 3-4, we alluded to the idea that ANGIE can support the word "plays" because all the requisite substructures are in place. This is also true for the FST because all the columns and column transitions required to form the parse for "plays" have been instantiated in *days* and *place*. When the FST is augmented with the back-off states, even more novel paths will exist thoughout the FST to support novel sequences. These permissible sequences correspond with some parse tree in the original ANGIE, and the parse score is distributed throughout the weights on the FST. From the output symbols, the novel letter-phoneme sequence can be extracted and both pronunciation and spelling can be deduced. That is, we can propose the underlying phonemic baseform as well as some letter-spelling hypothesis. By examining the morph class labels, partial information from the underlying parse tree can be recovered. This scheme is highly suitable for handling unknown words.

Morph	prince+	:	pr= =ince+
Onset	pr=	:	*p! r*
Rhyme	=ince	:	*i_x+ n ce*

Table 6-3: *Example of a Morph and its Decomposition into the Onset and Rhyme. Illustrated is the stressed root "prince+." An appended "=" annotates an onset and a prepended "=" annotates a rhyme.*

6.5 A Revised Two-Stage Architecture

At this point, let us revisit the two-stage architecture which consists of the same components as with our previous system in Chapter 5, with some modifications to incorporate the column bigram FST trained on a new letter-phoneme grammar.

6.5.1 Stage One

In stage one, the FST-based morph-level recognizer uses the same context-dependent acoustic models. Recall that the aim is to compute the precomposed FST U prior to recognition time:

$$U = C \circ P \circ L \circ G$$

Under the current architecture, the column bigram FST corresponds with P, transducing phones to letter-phonemes. A lexical FST L will map letter-phoneme strings to a set of lexical units. And G imposes n-gram model constraints on these units.

It must be underlined that emitting just letter-phoneme sequences at the FST has allowed greater flexibility for us in designing a set of lexical units. Defined in terms of these letter-phonemes and morph class labels, the lexicon is not required to coincide with the ANGIE lexicon used during training[7]. In the meantime, we begin by investigating the original set of morphs used in the ANGIE lexicon. Our intuition is that the stressed roots are characteristically specific to the fixed vocabulary, and do not generalize well to unknown words. In other words, stressed roots of unknowns are unlikely to be supported sufficiently by the current limited set of morphs. From an alternative viewpoint, many stressed roots are also sparsely supported in the training data. From a training set of over 47k sentences, more than half of the stressed roots are associated with 20 or fewer tokens. Stressed roots

[7] This point will be further reinforced in our discussions in the next chapter.

b=	bl=	br=	bu=	c=	cc=
ch=	chr=	cl=	cr=	cs=	cz=
dj=	dr=	f=	fl=	fr=	g=
ge=	gh=	gi=	gl=	gr=	gu=
h=	ju=	k=	kn=	kr=	ku=
l=	lh=	ll=	m=	mm=	n=
o=	ph=	phn=	phr=	pp=	ppr=
py=	qu=	quˆ =	r=	rh=	rr=
s=	sc=	sch=	scr=	sh=	shr=
sk=	sl=	sm=	sn=	sp=	sph=
spr=	squˆ =	ss=	st=	str=	su=
sw=	t=	th=	thr=	tr=	tw=
vl=	wh=	wr=	z=	zh=	zu=

Table 6-4: *Complete List of Syllable Onsets Decomposed from Stressed Roots. (An appending "=" denotes the stressed syllable onset.)*

also constitute the majority morpheme category in the morph lexicon (numbering 1259 units out of 1927). One measure is to decompose the original set of stressed roots into their respective onsets and rhymes, while keeping the original set of morphs that fall in the lexically unstressed categories. The result is a lexicon of morph and *submorph* units. From here on, we refer to this as the morph lexicon, although, strictly speaking, the set contains sub-syllabic units. An example of the morph decomposition is given in Table 6-3. In breaking up the set of stressed roots, the number of unique morph units is greatly reduced from 1927 to 1213 units. The set of stressed roots is collapsed to a set of onsets and rhymes. There are 545 items only, of which 455 are rhymes. These are also much better supported by the training data. The full set of onsets are given in Table 6-4 and the set of rhymes are given in Table 6-5.

Splitting into a set of onsets and rhymes and licensing novel sequences of these in the linguistic models will greatly improve flexibility and generalizing ability compared with using the original stressed root units. With 90 onsets and 455 rhymes, over 40k unique stressed roots can be synthesized by their concatenation, and this is supported in the smoothed linguistic models, in both the column bigram P and the trigram G. However, this does translate to a trade-off in the constraints by using more general units in the lexicon. But we hope that the degradation is moderated by the selective process in which we engineer more flexibility while cautiously maintaining much of the constraint.

=ˆ once+	=ˆ one+	=a+	=aalt+	=aam+	=ab+	=ac+	=ace+
=ack+	=act+	=ad+	=add+	=ade+	=adh+	=af+	=aff+
=ag+	=agh+	=ague+	=ah+	=ahr+	=ai+	=aign+	=ail+
=ain+	=aine+	=aint+	=air+	=ait+	=aj+	=ak+	=ake+
=al+	=ale+	=alis+	=alk+	=all+	=alm+	=alt+	=alz+
=am+	=amb+	=ame+	=an+	=anc+	=ance+	=anch+	=and+
=ane+	=ang+	=ank+	=ann+	=annes+	=ans+	=ant+	=ao+
=aos+	=ap+	=ape+	=app+	=aq+	=ar+	=arb+	=arc+
=arch+	=arck+	=ard+	=are+	=arge+	=arl+	=arm+	=arn+
=arr+	=art+	=as+	=ase+	=ash+	=ashe+	=ask+	=ass+
=at+	=atch+	=ate+	=ath+	=ati+	=att+	=atte+	=au+
=auck+	=aud+	=aul+	=aus+	=ause+	=ave+	=aw+	=awk+
=awr+	=ax+	=ay+	=aych+	=az+	=azz+	=e+	=each+
=ead+	=eak+	=eal+	=ean+	=eant+	=eard+	=earth+	=ease+
=east+	=eat+	=eath+	=eau+	=eaux+	=eav+	=eb+	=ec+
=eci+	=eck+	=ect+	=ed+	=edne+	=ee+	=eece+	=eed+
=eek+	=eel+	=een+	=eep+	=ees+	=eet+	=eeze+	=ef+
=eg+	=ei+	=eid+	=eight+	=eim+	=eir+	=eit+	=eive+
=eke+	=el+	=elf+	=elh+	=ell+	=elle+	=elp+	=elph+
=else+	=elve+	=em+	=eme+	=emp+	=en+	=ench+	=end+
=ene+	=eng+	=enh+	=enne+	=ent+	=eop+	=eoul+	=ept+
=er+	=ere+	=erg+	=erke+	=erm+	=ern+	=erre+	=erse+
=erst+	=ert+	=erz+	=es+	=ese+	=esh+	=esque+	=est+
=et+	=ete+	=eth+	=ett+	=ette+	=eur+	=ev+	=eve+
=ew+	=ex+	=ext+	=ey+	=eyk+	=hou+	=i+	=ib+
=ibb+	=ic+	=ice+	=ich+	=ici+	=ict+	=id+	=idd+
=ide+	=idge+	=iew+	=if+	=ife+	=iff+	=ig+	=igh+
=ight+	=igi+	=ign+	=ik+	=ike+	=il+	=ile+	=ill+
=ille+	=ilt+	=im+	=ime+	=in+	=inc+	=ince+	=inch+
=ine+	=ing+	=inh+	=ink+	=inn+	=insk+	=inst+	=int+
=ioux+	=ip+	=ipp+	=ique+	=irm+	=is+	=ise+	=ish+
=isle+	=iss+	=ist+	=it+	=ite+	=ith+	=iti+	=itts+
=iv+	=ive+	=ix+	=ize+	=izz+	=o+	=oa+	=oad+
=oak+	=oast+	=oat+	=ob+	=oc+	=och+	=ock+	=ode+
=odge+	=oe+	=oeur+	=of+	=off+	=og+	=ogg+	=ogne+
=oh+	=ohn+	=oi+	=oines+	=oint+	=ois+	=oit+	=oix+
=ok+	=old+	=ole+	=oll+	=olm+	=om+	=on+	=one+
=ong+	=onn+	=ont+	=oo+	=ook+	=ool+	=oom+	=oor+
=oot+	=op+	=ope+	=or+	=orces+	=ord+	=ore+	=orf+
=orge+	=ork+	=orld+	=orm+	=orn+	=orp+	=orr+	=ors+
=orse+	=ort+	=orth+	=orthe+	=orts+	=os+	=ose+	=oss+
=ost+	=ot+	=oth+	=ott+	=ou+	=oub+	=ouge+	=ough+
=ought+	=oul+	=ould+	=oun+	=ound+	=ount+	=oup+	=oupe+
=ourg+	=ourne+	=ourse+	=ous+	=ouse+	=out+	=outh+	=outhe+
=ove+	=ow+	=oward+	=owl+	=own+	=ox+	=oze+	=r+
=re+	=u+	=ub+	=uch+	=uck+	=ude+	=ue+	=uer+
=ues+	=uff+	=ul+	=ulf+	=ull+	=um+	=un+	=unc+
=und+	=une+	=unn+	=uns+	=up+	=uque+	=urch+	=urf+
=urg+	=urgh+	=url+	=urr+	=urs+	=urt+	=us+	=use+
=uss+	=ust+	=ut+	=utch+	=utt+	=ux+	=uz+	=uzz+
=ym+	=yp+	=yr+	=ys+	=yu+			

Table 6-5: *Complete List of Syllable Rhymes Decomposed from Stressed Roots. (A prepending "=" denotes the stressed rhyme.)*

In summary, following is a concise outline of linguistic constraints interleaved within the final composed FST structure:

- *Phonological and Phonotactic constraints:* These are embedded in models captured by the ANGIE-FST.
- *Grapheme constraints:* Knowledge derived from spelling information is captured in the ANGIE models in conjunction with sublexical phenomena and transformed into the ANGIE-FST.
- *Morph-level constraints:* Longer distance constraints are partially encoded by the ANGIE-FST which is trained from a word-level ANGIE grammar. These are also encoded in the trigram language model on the morph and submorph units.

6.5.2 Stage Two

As before, in the second stage, the search space is constrained by the phonetic networks output by the first stage. The arc weights on these networks consist of acoustic scores and language model scores. One modification is that a reduced weighting is imposed on language model scores. This scaling is determined on a held-out development set. This should aid performance, by enabling us to place less importance on the first-stage language models after the search space has been pruned, particularly as the second stage has the benefit of more powerful full word-level models.

As described in Chapter 5, the control strategy integrates together a word bigram and the ANGIE sublexical models, with the option of adding NL via the TINA module. We will not activate the NL in the experiments described in this chapter. We have modified the algorithm into a best-first search augmented with future estimates. These are potentials computed during the first stage Viterbi search, on the FST nodes. These alterations on the search are responsible for significant gains in running speed.

6.6 Recognition Experiments

The following experiments have been undertaken with a larger set of training data (which became available during the course of our research), and an expanded JUPITER vocabulary. The new JUPITER domain contains 1957 words encompassing 650 cities and 166 countries.

FST	Number of Arcs	Number of States
P without back-off smoothing	10175	2167
P with back-off smoothing	12093	2175
U without back-off smoothing	6.1m	341k
U with back-off smoothing	10.2m	690k

Table 6-6: *A Comparison of FST sizes, using the Column Bigram as the ANGIE-FST. FSTs with and without smoothing are included. P represents the ANGIE-FST with ANGIE derived probabilities as the arc weights. U represents a final composed FST embedded all language model constraints.*

All models are retrained using more than 47k utterances, a four-fold increase from our previous experiments.

For the ANGIE phoneme-based grammar retrained over the new data, there are 115 phoneme categories. This grammar is used in the second dynamic parse mechanism. For the ANGIE-FST, the letter-phoneme grammar is used. There are 289 letter-phoneme categories in total. On an independent test set of 1806 utterances, the per phone perplexity is computed using the phoneme grammar and the letter-phoneme grammar. The result is a 7.0% reduction in perplexity from 5.7 to 5.3. We anticipate that the reduction in perplexity will benefit recognition performance.

The FST compositions described in this chapter are implemented for JUPITER. The resultant FST sizes for the column bigram ANGIE-FST with and without the back-off mechanism are tabulated in Table 6-6. Also the final composed FST, U, is given, with and without the smoothing in the column bigram.

Observe that implementing a back-off mechanism increases the column bigram FST significantly, but the increase is even more pronounced in the final composed FST U. In our optimization procedures, determinization was unsuccessful on the final FST with back-off implemented, due to an explosion in the storage memory requirements demanded by the determinization algorithm for this FST[8]. In this case, the undeterminized FST, with large numbers of redundant paths intact, has a detrimental impact on the running time of the first-stage recognition search. Hence, for now, we abandon the back-off mechanism in these experiments but will return to it in the next chapter. The recognition experiments below

[8] Note that inherently, the FST *is* determinizable but its size and complexity at this stage precluded a successful implementation without consuming all the available memory storage. Our optimization schemes were conducted on a single 500MHz Pentium III machine with 1 gigabyte of memory.

System	WER(%)	UER(%)
Single-stage SUMMIT with trigram	8.0	8.8
Single-stage SUMMIT with bigram	9.2	11.3
Two-stage system	9.4	10.9

Table 6-7: *Performance on In-Vocabulary* JUPITER *data. Word (WER) and understanding (UER) error rates (%) are quoted for an independent test set of 1806 utterances, comparing two single-stage* SUMMIT *baseline systems with bigram and trigram language models and the two-stage system described in this chapter.*

will utilize a final pre-computed FST U which is composed from an ANGIE-FST P without the use of any back-off smoothing at the morph boundaries. Here, a limited determinization was possible; that is, an upper bound (of 8) on the number of arcs emanating from a node with identical input labels was placed. Optimal search speeds are not achieved owing to the lack of a fully determinized FST but, currently, running times remain satisfactorily near-real-time. We return to address this in the next chapter.

Recognition performance is ascertained from an in-vocabulary test set containing 1806 sentences. At this stage, our intention is to confirm the feasibility of the current two-stage configuration, which possesses much greater flexibility for supporting unknown words. It is compared with a state-of-the-art single-stage baseline which uses the same acoustic models and a word bigram and trigram. The baseline system is similar to that described in previous chapters. Our hope is that our two-stage system produces comparable results to the baseline, which has been optimized on in-vocabulary data, and establish that the flexible linguistic models can handle in-vocabulary data competently.

Performance results are shown in Table 6-7. As before, evaluation is conducted on an understanding error rate (UER) as well as word error rate (WER). The results show that the two-stage system performs most comparably with the single-stage baseline which applies a word bigram, and the performance loss remains small on both WER and UER. This is consistent, as the two-stage system here does not embed word trigram or NL knowledge, but only a word bigram in the second stage. A small degradation in performance is expected, as our scheme, compared with that in Chapter 5, is markedly more flexible. We have abandoned the multi-word units as well as full stressed root morphs, for a more general lexicon, resulting in less constrained language models. Underlyingly, the ANGIE-FST differs from the right-to-left branching method in relying less on memorizing training data, but rather, it emulates ANGIE's ability to predict unseen data. By comparison, the baseline system only performs marginally better here, whereas its accuracy plunges to below 50% on OOV sentences[9]. In conclusion, we are confident that we have arrived at a better position to cope with sparse data and rare pronunciations as well as tackle flexible vocabulary.

6.7 Final Remarks

At the conclusion of the last chapter, we established that the ANGIE probability model can be effectively encapsulated in an FST structure and deliver superior performance in the two-stage architecture. Our system exhibited favorable results but we identified some attributes that were divergent with our penultimate goals for a dynamic and flexible vocabulary system. This chapter has implemented a number of advances in the first stage in an attempt to progress towards that direction. The features introduced here bring us closer to a more general domain-independent first stage, with generic linguistic knowledge. The basis of these ideas has centered around the use of the ANGIE framework and FSTs to weave together disparate sources of linguistic information.

At this juncture, experimental results described above merely verified competence on an in-vocabulary test set. Naturally, the next step is to demonstrate an ability to handle data with OOV words. However, one stumbling block remaining is the implementation of an optimized and compact FST. As alluded to earlier, we encountered problems when composing the ANGIE-FST augmented with back-off states with the language model FSTs. The root of the problem lies at the combinatorial expansion of possible paths when a smoothing mechanism is incorporated, causing the optimization procedures to fail, under reasonable memory and computational resources.

Evidently, disabling the back-off mechanism is not detrimental to the recognition of in-vocabulary sentences because these sequences are well-supported during the training process. But the back-off mechanism would play a more important role in supporting unknown data. Hence, we must strive to conceive of a solution that affords an even greater degree of efficiency without recourse to alternatives that undermine the current level of flexibility and generality, achieved here. In the next chapter, we adopt a novel strategy for combating this issue, whereby a single algorithm can reap dual gains of improved probability

[9] This will be further discussed in Chapter 8.

modeling and memory efficiency, over and above the existing FST scheme.

Chapter 7

Automatic Lexical Generation

7.1 Overview

In this chapter, in addressing the issues of the current first stage, we describe the development of a new breakthrough approach: automatic lexical generation. We will explore the motivation and conception of optimizing the lexical space, and entertain the possible advantages. Following this, we present an iterative procedure that builds upon marrying together technological elements introduced in Chapter 6: the letter-phoneme grammar and the column bigram FST. The results of implementing the iterative algorithm are examined, while the novel lexical units are analyzed in depth. Finally, we will discuss the consequences of utilizing these in our first stage.

7.2 Motivation

With the advent of the column bigram FST capturing ANGIE's power to generalize, and the expansion of the ANGIE grammar to embody grapheme information, the need for greater compactness and efficiency in combining these linguistic models persists. This is ever more pressing in light of our vision to eventually train on larger and more general vocabulary corpora. And this is only possible in the first stage if the FST data structure achieves sufficient memory efficiency to enable a tenable recognition search speed to take place. As we intimated at the conclusion of Chapter 6, the pre-composed final FST is encumbered by a multitude of arcs and extraneous outputs. In order to attain maximal coverage and enhance contextual information, in comparison with the previous left-to-right branching method, the current FST has grown in size due to a host of new features. First of all, the new letter-phoneme grammar has more than doubled the size of the pre-terminal layer, thereby significantly expanding the ANGIE probability models. As the column bigram covers all adjacent pairs of ANGIE columns that have instantiated, coupled with the new letter-phoneme grammar, many more arc transitions are licensed. Extraneous nodes are created to indicate the position of morph and sub-morph boundaries. Although, the decomposition of stressed roots into onset and rhyme constituents has resulted in smaller sized trigram models, the smoothing mechanism that supports all combinations of (sub-)morph sequences (with all their associated pronunciation variants) opens many new pathways in the final FST, beyond practical proportions. It is now incumbent upon us to seek a solution without compromising the delicately interleaved linguistic constraints. In other words, it is undesirable to resort to looser forms of constraints, and forego the current lexical organization that, we surmise, is well-suited for OOV modeling as well as in-vocabulary recognition.

Let us ponder further the nature of our modeling approach that exploits probabilistic learning over symbolic representations that are fused with expert linguistic knowledge. This symbol set is seeded from a process whereby our intuitive knowledge is used to discern the most valuable information for our model constraints. Subsequently, the ANGIE models and letter-phoneme set are crafted and fine-tuned as best as possible to improve probability modeling. This is intrinsically a labor-intensive process, heuristic in nature, involving trial-and-error procedures and close analyses of data over time. What could benefit from this modeling philosophy is the application of automatic learning, optimizing on the probability likelihood while circumventing sparse data problems. During our experience, it became apparent to us that optimization is possible and indeed desirable at the lexical level. The organization of the lexical space in the first stage can be re-optimized. We explicate this concept in the next section.

7.3 Lexical Space Optimization

Core to the idea of re-optimizing the lexicon of morphs and sub-morphs is the fundamental insight that these morph units are not required to directly correspond with the word units of the later stages. In our prior systems, each word lexical unit of the second stage can be broken down to a sequence of constituents each of which appears exactly in the first-stage lexicon. But this is not strictly necessary. The reason is that the first stage decomposes all the morph hypotheses back to their phones to form a phonetic network, thereby abandoning those original morphs entirely. Our goal then is merely to maximize the quality of the phonetic network. The underlying morph units are free to be extracted or newly created from other means, as long as, ultimately, the phones that are eventually proposed by the first stage can formulate into the JUPITER words from the spoken utterance. Consequently, the words need not be associated with a single unique syllabification nor do the intermittent word boundaries between phones, dictated by the orthography, need to be preserved, because a different syllabification can be selected in the second stage when higher-level information becomes freely available. We posit that our chances for producing correct phonetic sequences for both known and unknown data may improve upon implementing some optimization on the lexical space whereby the language model probability likelihoods improve, yielding better perplexity measurements. It is our hope that, in the process, the overall models become more compact.

This leads us to pose the question of how to creatively manipulate the lexicon to improve perplexity. The key is to improve the ANGIE grammar by redefining the lexicon that implicitly trains the ANGIE models. By our definition, this also coincides with the lexicon on which the first-stage units and trigram models are based. An iterative algorithm has been conceived where a novel set of lexical units and a new ANGIE grammar are generated at each iteration. As in training ANGIE, the procedure uses forced aligned training data. The mechanics of the procedure hinges upon the following key features of the letter-phoneme column bigram ANGIE-FST:

1. Given any phonetic sequence, multiple paths can be found within the FST, emitting distinct letter-phoneme sequences at the output. This is attributed to the column bigram FST's over-generalizing property, enhanced with the back-off mechanism in place to counter missing or incomplete data. As a result, many novel output sequences in addition to the "correct" one co-exist, all accepting the phonetic sequence that was actually realized. These competing letter-phoneme outputs correspond to alternative ANGIE parses for the entire sentence, when restrictions to the underlying word baseforms are lifted and the original word boundary locations no longer apply. The cumulative FST scores correspond exactly to the scores computed by an ANGIE parser, with the exception of where the smoothing mechanism has been applied. It became

apparent then that among all the competing paths, the highest scoring one would not necessarily produce a letter-phoneme sequence that exactly matches the morphs asserted in the original orthography. But in fact, the letter-phonemes are juxtaposed to form novel alternate morphs. Under these circumstances, an alternative set of lexical units yields higher probabilities, translating to reduced perplexity measurements.

2. At the FST output, the letter-phoneme sequences and peripheral information can be pieced together to derive novel morphs. When stripped of their diacritics, the letter-phonemes concatenate to form novel spellings, offering a potential orthographic transcription. In fact, the operation of the parse mechanism is simulated via the FST insofar as information regarding the underlying parse can be ascertained from the FST output labels. We deduce the placement of morph boundaries along with the underlying morph class.

Hence, we shall develop an iterative procedure where, at every iteration, we seek the most likely paths through the ANGIE-FST given the phonetic realizations of the training data, arriving at a new lexicon each time. We emphasize that the lexical space will be optimized but the phonetic space remains a constant. The procedure does not engage the acoustic models, assuming that these are accurate, producing high quality alignments[1]. Under this assumption, the lexicon is optimized to facilitate the linguistic models in order to predict the given phonetic realizations.

7.4 The Iterative Procedure

Here, we provide a full exposition of the iterative algorithm. At the beginning of each iteration, an ANGIE grammar is trained from the current word and morph lexicon and phonetic alignments. The column bigram FST is then compiled. The next pass involves seeking the highest scoring path in the FST for the phonetic sequence of each training utterance. Subsequently, the FST outputs are collected. At this point, the new set of morphs and words must be determined; these are necessary to train up the ANGIE grammar for the next iteration. The morph spellings are inferred from the letter-phonemes themselves, and the morph class label outputs indicate the position of the morph boundary and the

[1] This assumption is flawed inasmuch as the acoustic alignments may suffer small errors. It is possible that these limitations in the alignments may adversely affect our results.

appropriate annotations to append to the morph. As for the formulation of words, these are not mandated by the FST outputs. The morph sequences can be creatively combined in multiple ways to form novel words of any numbers of syllables. For the sake of simplicity, we impose some empirical rules for concatenating morph sequences into new words. These will restrict the words to contain at most one stressed root or one function morph, in addition to conformance with ANGIE's original rules on ordering. For instance, a word boundary must be placed before a PREFIX morph, ensuring that the morph is in syllable-initial position. At the end of this pass through the training data, all the words and morphs are compiled into the two-tier ANGIE lexicons. The next step is to generate a new ANGIE grammar via retraining with the new lexicons. Note that the context-free rules remain unchanged throughout. And the next iteration can proceed with the new ANGIE grammar in the same way. A concise step-by-step summary is listed below:

1. **Initialization**:

   Begin with an initial set of rules that incorporate letter-phoneme units.

2. **Train grammar:**

   Use the forced aligned set of orthographic and phonetic transcriptions to train an ANGIE grammar.

3. **FST Generation:**

   Use the forced alignments and trained ANGIE grammar to generate a column-bigram FST.

4. **Search:**

   For the phonetic sequence of each training utterance, use a best-first search to find the highest scoring path through the column-bigram FST, and output the corresponding letter-phoneme sequence along with morph class labels at morph boundaries.

5. **Construct morphs:**

   For each morph, infer the spelling by concatenating the letter-phoneme sequence, after removing contextual markers. The morph class is deduced and the relevant diacritic is added. For example, if the FST output consisted of the sequence, *d!* ONSET *ay+*

RHYME, we can elicit that the underlying morph is the stressed root, *day+*. If this morph has not been previously encountered, add it to the lexicon.

6. **Construct words:**

   Construct the underlying "word" by concatenating morph sequences using some simple rules. Each word must contain only one stressed morph, and word boundaries are inserted whenever permissible according to ANGIE. For example, if *day+* and *=s* appear in sequence, these can be concatenated together to form the word *days*. If the word has not been previously encountered, add it to the lexicon.

7. **Go to step 2:**

   Upon completion with compilation of the new lexicons, generate a new alignment file with the *same* phonetic sequences but with new orthographies. We can now return to Step 2 to train a new ANGIE grammar.

At every iteration, the probability likelihoods improve by some amount, because each time, the highest scoring path in the FST is sought for the same phonetic sequences. Let us highlight that searching for alternative viable paths for each individual sentence is entirely possible owing to the ANGIE-FST, which encapsulates the space of alternate parses. Yet this is done under a limited search space. Attempting to search through all alternate parses using the dynamic parse mechanism for each sentence would be quite formidable.

7.5 Results of Iteration

The above algorithm is implemented with the JUPITER corpus. It was found that after four iterations, the ANGIE grammar converges (in the sense that both the word and morph lexicons did not change further.)

In the final converged grammar, the total lexicon size actually grew. The number of words increased from 1957 to 3516; the number of morphs increased from 1927 to 2071. Prior to iteration, there were 1259 stressed roots, but this number rose to 1589 after the procedure. Thus, the percentage of stressed roots in the morph lexicon increased from 65.3% to 76.6%. With the increase in the ANGIE word inventory, multiple mappings exist from the lexicon to each underlying spoken word. At each distinct phonetic instantiation, the highest scoring path in the ANGIE-FST may be unique, yielding a different parse with

Grammar	Arcs	States
Original Phoneme	9385	1488
Letter-phoneme	12k	2175
Final Iterated	9741	1717

Table 7-1: *Size of FSTs with Different ANGIE Grammars.*

different higher-level morph or word structures[2]. In examining the search more closely, we discovered that many preferred paths visited the back-off mechanism, whose role is critical in enabling many novel sequences with favorable probability estimates to be produced.

While the number of lexical units expanded, the actual ANGIE grammar size progressively shrank at every iteration. The number of unique columns in ANGIE decreased from 1439 to 1164. At each iteration, in searching for the highest scoring paths for all training sentences, some portion of the probability space in the ANGIE-FST is never visited even though that space has been allocated by the grammar. In a sense, these uncharted paths in the FST are unnecessary, and are to be eliminated in the following iteration. We interpret this to suggest that a smaller ANGIE-FST, stemming from a smaller ANGIE grammar, is sufficient to model the same phonetic space. By convergence, the number of letter-phonemes in use from the original inventory is 264 (from 289). This can be viewed as an automatic elimination of the ones that were sparse, producing unlikely probabilities. In Table 7-1, the sizes of the column bigram FSTs for different grammars are displayed. The final iterated ANGIE-FST exhibits an 18.8% reduction in the number of arcs and a 21.1% reduction in the number of states, in comparison with the pre-iterated grammar. The final size of this column bigram letter-phoneme based FST is marginally larger than one that is derived from a phoneme-based grammar.

When the stressed roots are decomposed into their constituents, there is only a total of 418 onset and rhymes, compared with the 545-sized prior set. This constitutes a drastic reduction. Along with another 482 unstressed morphs, there is a total of only 900 units in the lexical FST and the trigram models. Compare this with 1213 units of the original pre-iterated grammar.

When the ANGIE-FST is composed with the lexical and trigram FSTs, the final optimized

[2] We examine the lexical units in greater detail in the next section.

FST U is fully determinized. As a result, U contains 4.3 million arcs and 440,000 states, occupying 90 megabytes of memory. This is less than half the size of the full pre-composed FST using a pre-iterated ANGIE-FST.

7.5.1 Perplexity Measurements

One way to gauge the success of the algorithm is to measure perplexity on the ANGIE grammar[3]. Again, per phone perplexity is measured on the same 1806-utterance test set. We witness a steady fall at every iteration. The final perplexity number is 4.9. This is a 7.5% reduction from the pre-iterated grammar which measured 5.3. Compared with the phoneme-based grammar whose perplexity is 5.7, there is a 14% relative perplexity reduction.

Another computation that may provide some insight on the effect of this algorithm is to closely examine the total probability space in the ANGIE-FST that has been visited. These results are tabulated in Table 7-2. First of all, we consider the likelihood score on all the FST arcs. An average is computed after weighting the numbers with the frequencies that the arc transitions were taken, during a pass through the training data. As expected, it is discovered that this weighted average increases successively comparing the phoneme-based ANGIE-FST, the letter-phoneme ANGIE-FST and the iterated ANGIE-FST. Furthermore, we consider the sum of probability on the arcs exiting each node in the FST. Again, we compute a weighted average based on the frequencies that the arc transitions were taken. The results also exhibit a small increase with the final iterated ANGIE-FST. Like perplexity, these numbers may suggest that the new iterated grammar better predicts the data. Whether all these positive results translate to improved recognition performance will be seen in the next chapter.

7.5.2 The Novel Lexicon

Let us conduct an in-depth perusal of the nature of the novel units from the final iterated grammar. A portion of the morphs and words from the original lexicons are preserved in the final iterated lexicon. 630 words (32.2% of the original inventory) remain unchanged,

[3] Note that the perplexity number is not an entirely precise measurement, because in ANGIE, a portion of the probability space has been allotted to ambiguous or alternate parses. These parses are not easily recoverable. Hence, the real perplexity is probably lower than what is attained by computing likelihoods from the single highest scoring parse.

Grammar	Weighted Average Score of Transitions	Weighted Average Sum of Probability Exiting Nodes
Original Phoneme	0.42	0.89
Letter-phoneme	0.47	0.89
Final Iterated	0.52	0.91

Table 7-2: *Probability Measurements of Column Bigram ANGIE-FSTs. Various ANGIE grammars are compared. See text for explanations.*

whereas 1087 morphs (56.4% of the original inventory) remain unchanged. Characteristics of the new units that have emerged are documented below.

- **Changes in Spelling:**

  Some words and morphs have changed in spelling because the algorithm preferred an alternative letter-phoneme sequence to the original designated one. Examples are listed below.

Original Words		Original Morphs	New Words		New Morphs
Beijing	⇒	bei+ jing+	bay ging	⇒	bay+ ging+
difference	⇒	diff+ er -ence	diference	⇒	dif+ er -ence
difference	⇒	diff+ er -ence	difurence	⇒	dif+ ur -ence
Edmonton	⇒	ed+ mon =ton	edmanton	⇒	ed+ man =ton
flying	⇒	fly+ =ing	flighing	⇒	fligh+ =ing
Kuwait	⇒	ku- wait+	kuwate	⇒	ku- wate+
London	⇒	lon+ -don	lundon	⇒	lun+ -don
marine	⇒	ma- rine+	mareen	⇒	ma- reen+
Ottawa	⇒	ott+ a -wa	audowa	⇒	aud+ ow -a
Sri_Lanka	⇒	sri- lank+ -a	shree lonca	⇒	shree+ lonca+
through	⇒	through+	throo	⇒	throo+

Table 7-3: *Examples of Modified Spellings for Words and their Morph Decompositions. Some words and their corresponding morphs before and after the application of our iterative procedure are shown.*

It is apparent that the algorithm selects a preferred spelling on the grounds that it better reflects the actual pronunciation, yielding higher probabilities. The particular realization may vary in spelling, depending on the phonetic pronunciation whose variations may be attributed both to different inherent ways to pronounce the word and/or phonological rules. Some of these amount to subtle differences in phonetic realization. This is exemplified in the different new spellings for the word *difference*:

diference, difurence. In one sense, this shifts the role of modeling alternate pronunciations into the level of the lexicon and alternate orthographies, by forcing the creation of new lexical units.

- **Changes in Syllable or Word Boundaries:**

  In many instances, the syllable or word boundaries of the original sentence have been altered. There exist several scenarios. Syllabification of some words can change where a consonant switches syllable affiliation, indicating that the new parse gives better probability estimates. One example is *Antarctica* shown below. For this particular case, it is interesting to note that this particular re-syllabification favors maximal onset over stress dominance.

Original Words		Original Morphs	New Words		New Morphs
Antarctica	⇒	an- tarct+ i -ca	Antarctica	⇒	an- tarc+ ti -ca

Table 7-4: *Example of Re-syllabification. The syllabification of the word Antarctica before and after the application of our iterative procedure is shown.*

At times, letter-phonemes that reside at word boundaries switch word affiliation, creating novel words. That is, individual consonants switch between the onset and coda position across word boundaries. Examples are given below.

Original Words		Original Morphs	New Words		New Morphs
ask thank	⇒	ask+ thank+	askth ank	⇒	ask+ =th ank+
for Quebec	⇒	for* quˆ e- bec+	fork abec	⇒	fork+ a- bec+
your name	⇒	your* name+	yorn ame	⇒	yorn+ ame+

Table 7-5: *Modification of Word Boundary Affiliations for Consonants. Examples of words and their corresponding morphs before and after the application of our iterative procedure are shown.*

Another related phenomenon occurs when entire morph units switch word affiliation intact. This may suggest that, for instance, a suffix of one word is better modeled as the prefix of the following. For example, the *-ta* suffix is converted to a *to-* prefix in *atlanta georgia* shown below. In *Ivory Coast*, the *-y* suffix of *Ivory* is re-spelled and transformed to the prefix *re-* for the next word. Evidently, *re-* is more commonly occurring.

Original Words		Original Morphs	New Words		New Morphs
Atlanta Georgia	⇒	at- lan+ -ta geor+ -gia	atlan togeorgia	⇒	at- lan+ to- geor+ -gia
Ivory Coast	⇒	i+ vor -y coast+	ive recoast	⇒	ive+ re- coast+

Table 7-6: *Modification of Word Boundary Modifications for Morphs. Examples of words and their corresponding morphs before and after the application of our iterative procedure are shown.*

Effectively, this data-driven method has chosen a more optimized syllabification in terms of likelihoods compared with the original, hand-crafted one. The learning in the algorithm winnows out those hand-selected units that yield poor modeling.

- **Foot-like Compound Units:**

  Some instances of novel word creation are exemplified by the clustering of several adjacent mono-syllabic words into new multi-syllabic ones. Under these circumstances, a single word associated with one of the syllables is assigned lexical stress, whereas the surrounding ones no longer contain lexical stress and are identified as prefixes or suffixes. Examples are listed below.

Original Words		Original Morphs	New Words		New Morphs
a good day	⇒	a* good+ day+	agoodday	⇒	a- good+ -day
to rain	⇒	to* rain+	torain	⇒	to- rain+
at Wimbledon	⇒	at* wim+ ble -don	atwimbleden	⇒	a- twim+ ble -den
I'm interested	⇒	iam* in+ ter -est =ed	aminterested	⇒	a- min+ ter -est =ed

Table 7-7: *Examples of Foot-like Compound Units Derived from Iterative Procedure.*

On close examination of these results, these new word units seemingly characterize the rhythmic properties of the sentence realization. In abandoning the original demarcation mandated by the underlying orthography, we have generated new ones that only account for phonetic, syllabic and stress patterns, albeit using a representation that utilizes creative spellings. Intuitively, focusing at the sentential level, these bear some semblance to rhythmic foot units, described in metrical phonology, [31, 40, 63] which studies the theory of stress and espouses the use of syllables and higher-level units to capture stress patterns and rhythms. The notion of constituents called rhythmic foot units was raised to characterize the alternating strong and weak stress patterns in spoken English that some argue contribute to a perceived regularity in rhythm. That is, at roughly equal intervals, it is possible to discern certain rhythmic beats that cause an "isochronal" movement, stemming from stress and prominence patterns. Although the exact definition of a rhythmic foot is under dispute, a stress foot is provisionally defined as a string containing as its first element a stressed syllable followed by zero or more unstressed syllables. It is also said that foot boundaries can coincide with prosodic boundaries. Most intriguingly, Giegerich [25] went so far as to suggest that the foot unit has influence on phonological processes. We argue that our novel units capture succinctly rhythmic patterns of the spoken input in the same way, and our units were automatically discovered via examining sublexical and phonological patterns in the data. More examples can be observed in the table below, where the original orthographies are compared against the new ones with novel spellings. In each case, every "pseudo-word" marks the location of a single stressed syllable.

Old:	Mineapolis
New:	miny apolis
Old:	Wyoming
New:	wai oming
Old:	January
New:	jan you ari
Old:	how about Dominican Republic
New:	howa boutda minne can republic
Old:	who created you
New:	who cree eightidge you
Old:	expected in Samoa
New:	expectidence amoi
Old:	I'd like to check another city
New:	id like tocheca nother city
Old:	the average relative humidity
New:	the ave rejanual relative humidity

Table 7-8: *More Examples of Sentences with Novel Words Compared With Their Original Orthographies.*

7.6 Final Remarks

We have enjoyed success in designing and implementing a novel algorithm that has served the dual purpose of reducing final FST size and improving probability likelihoods or perplexity. This is achieved without major compromises on the two crucial factors of our design: flexibility and linguistic constraint.

More interesting are some of the characteristics manifested by the final lexical units. These ranged from modifications in spelling to syllabification. The automatic algorithm appears to have learned the rhythmic properties of a sentence and synthesized units that express these patterns of alternating stress. We claim that the learning process has discovered a set of prosodically motivated linguistic units. This can be interpreted as another promising indicator, possibly leading to better recognition performance.

At this point, we have arrived at an innovative recognition engine that possesses many of our original criteria for a low-level generic first stage. The next step is to assemble a full multi-stage system, and investigate performance on handling flexible vocabulary. We seek answers to whether our system can salvage unknown words from queries, process them and propose possible spellings.

Chapter 8

Unknown Word Experiments

8.1 Introduction

In the preceding chapters, we have identified the criteria for a flexible domain-independent first stage, and have set out to satisfy these, using technologies based on ANGIE and FSTs. On the way, we have endeavored to infuse our design with the conflicting factors of generality and constraint. In Chapter 6, we have witnessed that a version of our first stage performs competently on sentences containing only in-vocabulary data. And, in the last chapter, we took further steps to improve the probability models by adopting an entirely novel lexicon. The previous developments have culminated to the point where we are now in a position to assemble a full system, and conduct experiments on data with OOV items. At the time of this experiment, the state-of-the-art system [110] in the JUPITER domain, performs at around 9.9% WER for a test set with only in-vocabulary words. This more than doubles to 19.1% for the entire test set, where OOV and out-of-domain sentences constitute over one quarter of the entire set. These sentences are the culprit for a large plunge in the overall accuracies. By themselves, the WER stands at 54.0%. For every utterance with an out-of-domain component, more than 3 errors are committed on average. This signifies that the system breaks down in the face of sentences outside of its range of handling, notwithstanding its superior accuracy on in-vocabulary data for which it has been tailored and optimized. We posit that significant gains may emerge for utterances containing some unknown words when our system is applied. Particularly, in detecting the unknowns, our system can salvage those OOV sentence that would have been otherwise misunderstood in their entirety. The result will somewhat narrow the large performance gap that exists between in-vocabulary and out-of-domain utterances.

In the following, we present a three-stage architecture that consists of some enhancements from the original two-stage concept. The original two-stage arrangement is subsumed as one mode of operation in the three-stage arrangement. For each stage, we will outline the components which have been drawn from the technologies developed in the previous chapters. Recognition experiments have been undertaken on a test set with OOV words. For the moment, we address the portion of our data that involves queries for unknown city names only. This is particularly apt, as in such queries, the system's usability can immediately benefit when a reference to an unknown city at the spoken input is detected. An appropriate response can be generated, but also, a hypothesis for the city name can be confirmed with the user, and the subsequent city can be used to extend the lexicon. The experiments quoted here will aim to recognize and understand queries regarding unknown cities. Secondly, we attempt to automatically extract spelling hypotheses for the unknown city names. We emphasize that the system has not been fine-tuned for its sound-to-letter capabilities. And the task of recognizing unknown words, their underlying phones and possible orthography is an ambitious feat. This will be a first attempt to investigate whether this is at all a possibility. Results will be ascertained for both a two-stage and a three-stage variant of our system configuration, comparing their respective merits. This chapter will conclude with a series of analyses and discussions about the results of our experiments, implications for our design and efficacy of the overall system.

8.2 Three-Stage System

In this new configuration, an additional search pass, designated for applying natural language models, is appended to the original two-stage architecture. This enables us to investigate the relative merits of applying above word-level linguistic models at an isolated final stage, in comparison with our current integrated scheme. The practicality of these two modes needs to be evaluated under the context of handling real data that contain OOV words interspersed with in-vocabulary words. We anticipated that, ideally, applying various sources of linguistic knowledge under a tightly coupled control strategy would empower the models to mutually interact in a way that would increase accuracy. Nonetheless, this may remain untenable, particularly when the search space is opened to accept unknown words and novel word constructions at any time. The alternative is to postpone the natural language models to the final stage.

Figure 8-1 depicts the three-stage architecture which we shall describe in this section. In essence, linguistic knowledge is applied from the bottom upwards as we successively proceed from one stage to the next. Acoustic knowledge drives the initial recognition pass, which is aided by the low-level largely domain-independent ANGIE-FST models that were developed in Chapters 6 and 7. The second pass primarily adds word-level information with the ANGIE parse mechanism. Its role is to determine possible locations for unknown words, and to propose in-domain hypotheses. Although one option is to apply natural language information here, a third stage is designated to traverse an even more slimmed down search space, enlisting TINA NL models to determine the final sentence hypothesis and meaning representation.

8.2.1 Stage One

In stage one, the acoustic model and search aspects of the recognition engine are as previously explained in Chapter 5. The FST recognizer loads in the precomputed and optimized FST U, as described in Chapter 7. In summary, let us highlight the major features in this FST, where $U = C \circ P \circ L \circ G$ (also depicted in Figure 8-1):

- P is a column bigram ANGIE-FST trained from a letter-phoneme based grammar that has been iterated to optimize on the lexical units.
- L transduces the letter-phonemes to the novel morph units, further decomposing the stressed roots into onsets and rhymes.
- G is a trigram FST on the novel morphs.

The recognizer outputs an optimized phonetic network whose arc weights consist of the acoustic scores and language model scores with reduced weighting.

8.2.2 Stage Two: The ANGIE-Based Search

The role of the second stage is to traverse the pruned search space defined by the phonetic network from stage one, and identify potential word hypotheses as well as possible locations of unknown words. As in the two-stage system explained in Chapter 6, the integrated search

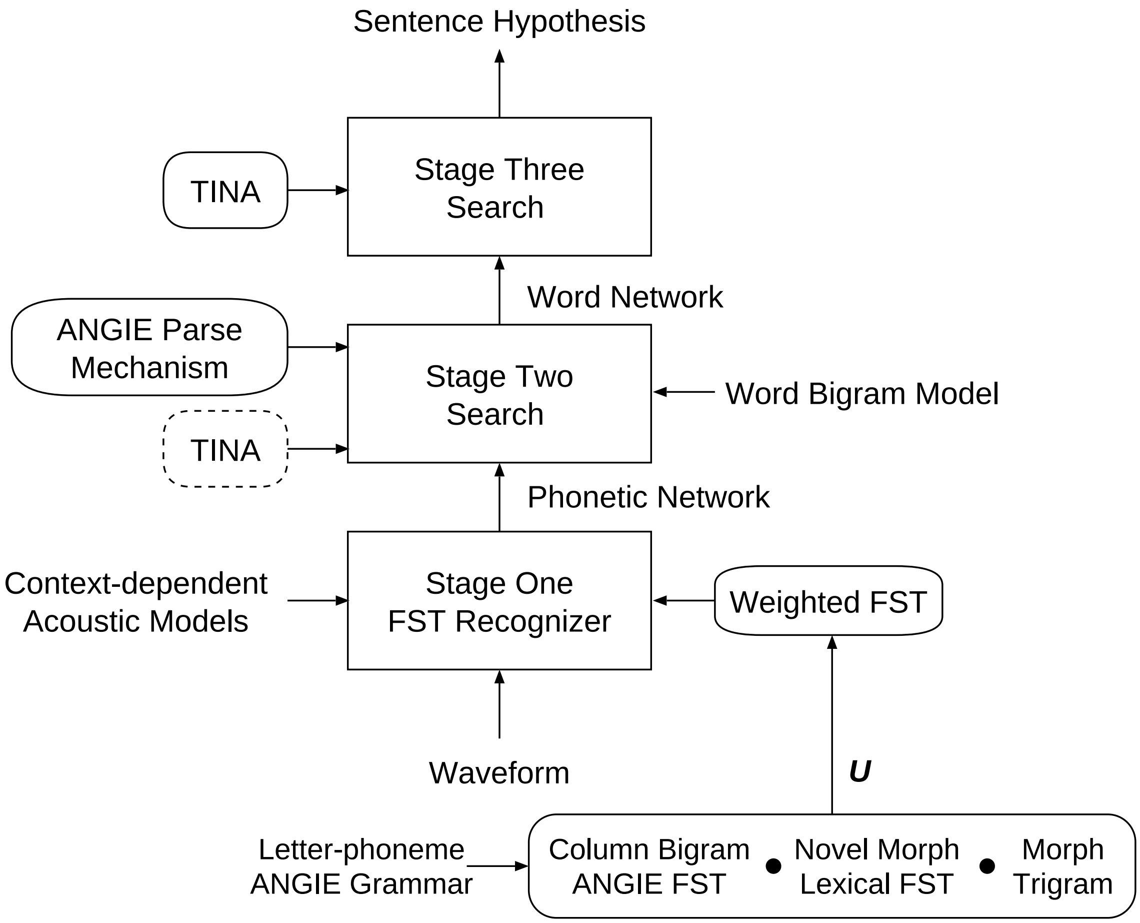

Figure 8-1: *Block diagram of Three-Stage System.*

strategy utilizes a best-first search using future estimates, coordinating the ANGIE dynamic parse mechanism, a word bigram and, optionally, TINA.

As in the previously described second stage, ANGIE seeks to discover words from the bottom-up along phonetic sequences. Generally, a phonetic sequence has to parse successfully, as stipulated by the context-free rules and probability models. When operating exclusively to support in-vocabulary words, the mechanism needs to access a lexicon associated with the pre-terminal (letter-phoneme or phoneme) layer of the parse tree and to find a matching entry. However, if OOV words are allowed, and the pre-terminal sequence of the proposed parse cannot be matched in the lexicon, an unknown word is proposed. ANGIE returns an OOV flag along with the current theory and score, which are returned to join the stack. Under this bottom-up scheme, all theories that contain word hypotheses with successful parse trees are pursued, and compete among one another. This prevents the phonetic sequences that score favorably in both the acoustics and the ANGIE parse from being pruned away prematurely. To counter the over-generation of unknown word hypotheses, and the ballooning of the search by too many competing paths, various measures have been erected.

While bigram scores are expected to discourage the proliferation of unknown word hypotheses, an empirically determined constant is used to penalize each unknown word hypothesis. As we only deal with utterances with unknown city names, we further impose the restriction that unknown words can only occur following a short list of words. These are found to be words preceding city names in the training data. They are provided here: *in, for, uh, oh, um, on, at, is, about, what_about, like.*

In preliminary investigations, we discovered that the tendency for ANGIE's flexible grammar to generate excessively long unknown words needs to be curbed. Therefore, each unknown word is restricted to only one stressed root and one derivational suffix. This disables the grammar's ability to recursively build single-word parses with prohibitively large numbers of syllables. In case where the unknown city is multi-syllabic with more than one stressed root, the grammar is forced to start a new word, incurring an additional unknown word penalty. At the output of stage two, multiple adjacent unknown word tags are collapsed into a single one. They are treated as a single unknown city name.

As previously described, when ANGIE locates a word boundary and returns the word identity, the higher-level language models are applied. Here, TINA can be optionally applied to parse the current string of words in the partial sentence hypothesis. TINA is trained to handle unknown words exclusively as unknown city names in the sentence. If other unknown words such as other proper names are to be handled, TINA could be extended to accept unknowns under other categories. We will descibe this further.

The second stage outputs a list of N-best scoring sentences. The unknown word hypotheses are tagged as <unknown> among the in-vocabulary hypotheses. However, it is also possible to extract spelling hypotheses for the unknown words. This necessitates the use of the letter-phoneme grammar in the ANGIE parse mechanism. Then, the sequence of pre-terminal letter-phonemes in the unknown word parse is retrieved. The annotations are stripped and letter-phonemes are concatenated to form a possible spelling. This word appears in the N-best list.

8.2.3 Stage Three: TINA Parsing

Instead of applying TINA in conjunction with ANGIE in stage two, our new scheme postpones NL processing to the final third stage. We expect this strategy to lighten the computational load in stage two, and yet, the third stage will also be efficient due to a highly pruned search space.

In an approximate algorithm[1], the N-best output of stage two is converted into a word network, collapsing words that occur in the same relative position of the sentence. Goodness scores for each word are computed by ranking hypotheses according to their frequency of occurrence in the list. Sweeping the network using a Viterbi search with beam pruning, TINA parsing, with the robust parse handling enabled, is applied. The highest scoring sentence, with TINA and the goodness scores combined, is sought by the search. Another benefit is that a meaning representation can be obtained directly during this stage, to be used for further dialog processing.

[1] Note that this approach was taken for a fast and simple implementation. The scores from the second stage are not retained, but, in principle, they can be combined with the TINA scores in the third stage.

8.3 Experiments

8.3.1 Training

56k sentences are made available for training language models. For the recognition experiments, a phoneme-based ANGIE grammar is used in the second stage ANGIE parser. This, being smaller than the letter-phoneme grammar, should ease computation during the search process. In order to train TINA to handle unknowns appropriately, we employed a heuristic approach. During training, one out of every ten training sentences containing city names is selected at random. These city names are replaced by an "unknown" tag so that in the next training pass, the TINA models treat these as unknown cities. Consequently, the probabilities for encountering unknown city names are artificially boosted. Within the 56k training sentences, there were approximately 2000 sentences that were artificially augmented with the "unknown" marker. We did not attempt to employ more sophisticated training, though, in actuality, system performance may benefit if the unknown city rate in the training is made to match closely to that of real usage.

A development set of 430 utterances was set aside for determining other parameters.

8.3.2 Recognition

In the recognition experiments, performance is evaluated on an independent test set of 425 utterances. This set is specially chosen such that all sentences pertain to weather information queries regarding unknown cities. Each test utterance contains exactly one unknown city name.

The baseline recognizer is a single-stage SUMMIT [110] recognizer which does not have the capability to handle OOV items. It uses the same context-dependent acoustic models as the three-stage system, and a bigram and trigram word model trained on the same training corpus. This baseline system has been described in previous chapters.

For comparison, experiments are conducted on three variations of our system. We consider:

- **System 1 Two-Stage ANGIE only**: A two-stage only version where the top scoring sentence hypothesis of the second stage is evaluated and NL processing is omitted.
- **System 2 Two-Stage ANGIE-TINA**: A two-stage version which employs both ANGIE and TINA in stage two.

Key	Example Values
TIME_OF_DAY	afternoon, morning
WEATHER	temperature, pollen count, windspeed, pressure, rain
CURRENT	yes
REGION	massachusetts
UNK_CITY	timbuktu
QUANTIFIER	average_rain, accumulation
CITY	boston
CLAUSE	time, info
CARDINAL	south, west
DATE	sunday, today
CRISIS_TYPE	hurricane flood

Table 8-1: *List of Concepts and Example Values for Understanding Evaluation.*

- **System 3 Three-Stage System**: the full-fledged three-stage system.

In the three-stage system (System 3), the third-stage word network is constructed from the N-best list of stage two with $N = 20$. Results are reported for word (WER) and understanding (UER) error rate. The latter is computed on a set of concept-value pairs. There are 11 concept types, tabulated in Table 8-1 along with example values for each. When an unknown city is detected in a parse, the UNK_CITY concept is proposed. A sentence is recognized/understood correctly if all the known words are recognized/understood, and an unknown flag is proposed at the correct location in the sentence where the unknown city name is spoken.

8.3.3 Results and Analysis

WER and UER for the baseline and the three experimental systems are illustrated graphically in Figure 8-2. The breakdown of the WER and UER numbers are tabulated in Tables 8-3 and 8-4.

The baseline system achieved a WER of 24.6% and UER of 67.0%. Upon closer examination, in spite of the incidence of exactly one unknown city per utterance, the system committed on average 1.9 recognition errors per utterance. This evidence reinforces our notion that the prevalence of unknowns multiplies the difficulties in recognizing the surrounding in-vocabulary words. Due to the complete absence of OOV handling ability, sentence error rates are 100%. That is, the system's responses to *all* of the test sentences

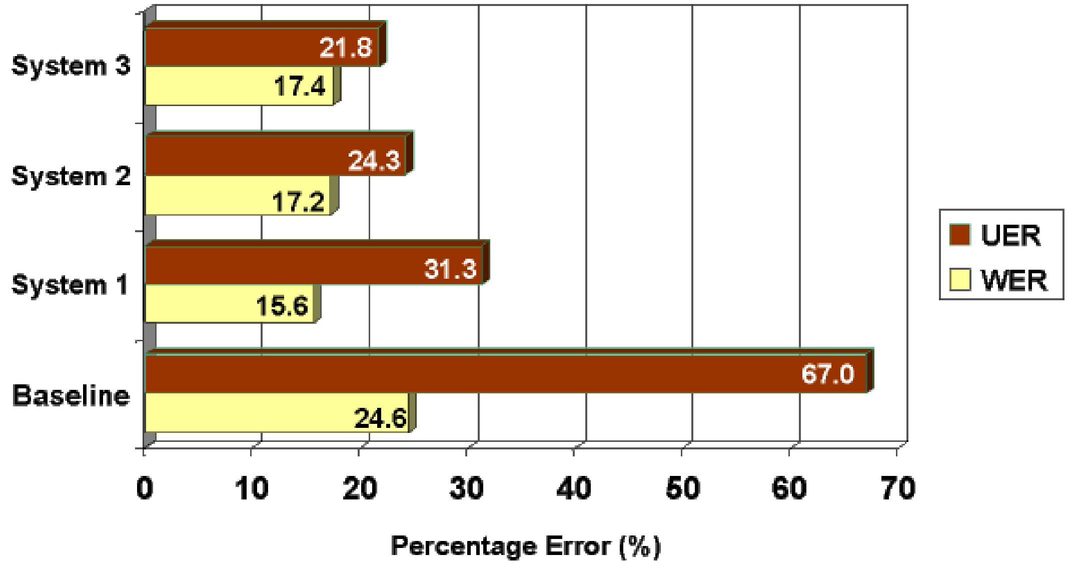

Figure 8-2: *Bar Graph of Word and Understanding Error Rates.*

would have been incorrect. This is true despite the UER of 67% where some concepts such as *weather* and *temperature* were understood correctly. Large deletions are incurred in the UER, as the unknown city category is never detected. Usually an in-vocabulary word is proposed in its place, incurring an insertion. Deletions are also caused by large numbers of parse failures in the understanding evaluation. 78 utterances or 18.35% of the test set failed as the recognizer tends to propose nonsensical words at the unknown spoken input. (See Table 8-2 for a comparion of the number of failures for each system.)

Significant improvements are made using System 1 with a WER of 15.6% (36.6% relative improvement) and UER of 31.3% (53.3% relative improvement). The two-stage system substantially reduces the number of substitutions (from 15.2% to 5.0%) in word errors, and the number of deletions in UER is almost halved. On average, there are 1.2 errors committed per utterance, compared to 1.9 in the baseline. We computed the unknown word detection error rate by seeking all the unknown words that were discovered in the correct location. It was found that a detection error rate of 21.2% stands. This is composed of 2.6% false alarms and 18.6% misses.

These results are encouraging, particularly because NL constraints have yet to be utilized. However, at this point, the number of failures in understanding evaluation remains high, at 44 or 10.35% of the test set. In these utterances, parse failures prevent the generation of a meaning representation. This translates to a total lack of understanding on the part of the system. We expect the utilization of NL constraints to favor utterances that are meaningful, thereby lifting performance on this subset.

For System 2, coupling TINA and ANGIE within the recognition search, while licensing the proposal of unknown words, has proven to be an expensive computational overhead. The time required to run a test set increased over tenfold for the second stage when compared with System 1. However, some gains in understanding are reaped, with UER at 24.3%. Compare this with 31.3% in System 1 and 67.0% in the baseline. The number of utterances that failed to parse is reduced to 13 (3.1% of the test set). As consistent with observations in previous experiments, the WER slightly degrades despite improved understanding, upon addition of NL models. Although these performance gains are significant, they were accompanied by an escalation in the running times. TINA was employed in full-parse mode only. This is suboptimal, as a realistic application would certainly require robust parse handling.

The best UER and WER performance was derived from System 3, the full-fledged three-stage system, employing TINA in robust parse mode in the final search pass[2]. The WER stands at 17.4% (29.3% relative improvement from baseline) and the UER stands at 21.8% (67.5% relative improvement from baseline). Major reductions in the number of substitutions, deletions and insertions resulted. And a total of 7 utterances (1.6% of the test set) failed in understanding evaluation. These substantial gains signify the efficacy of the three-stage architecture and the word network interface. On close analysis of the recognizer outputs, we observed some tendency for the recognizer to identify portions of in-vocabulary words following unknowns as OOV, thereby causing an increase in the number of in-vocabulary deletions. But the number of substitutions and insertions in the WER has plunged compared with the baseline. The computation time is markedly shorter than in System 2. See Section 8.3.5 for details.

The above systems have manifested substantial gains in WER and UER, particularly when compared with a baseline that is not equipped with OOV handling capability. By and large, it has been demonstrated that our strategy has achieved the balance of preserving high in-vocabulary recognition accuracy along with successful unknown word detection. These gains are reflected in the error rate numbers, but during a real dialog, our three-stage system can potentially greatly improve usability, in recovering from a situation where an unknown city has been queried. All this is accomplished at near-real-time speeds, comparable with

[2] TINA is omitted from the second stage here.

System	No. of Fails	% Fails
Baseline System	78	18.35
1. Two Stage	44	10.35
2. Two Stage with TINA	13	3.1
3. Three Stage	7	1.6

Table 8-2: *Comparison of Parse Failures During Evaluation for 4 Systems. Results given for the 425-utterance test set. A marked drop in the number of failures is observed for the systems that employ* TINA *(System 2 and 3).*

System	WER(%)	Subs (%)	Ins (%)	Del (%)	SER (%)
Baseline System	24.6	15.2	6.5	2.8	100
1. Two Stage	15.6	5.0	3.2	7.4	59.8
2. Two Stage with TINA	17.2	10.7	2.2	4.3	59.3
3. Three Stage	17.4	2.8	1.9	12.7	57.2

Table 8-3: *Breakdown of Word Error Rates (WER) for baseline system and three experimental systems on a 425-utterance test set with unknown city names.*

the baseline configuration. Section 8.3.5 further investigates this aspect.

8.3.4 Spelling Extraction

In a pilot experiment, we test the feasibility of instantaneously proposing letter spellings for new words. As the purpose is simply to investigate viability, we did not attempt to implement spelling extraction with the three-stage system. Instead, a two-stage architecture is used, omitting the application of TINA. The letter-phoneme grammar is used in the second-stage ANGIE parser. As this is substantially larger than the original phoneme-based grammar, the search computation is accordingly increased. When a final sentence hypothe-

System	UER(%)	Subs (%)	Ins (%)	Del (%)	SER (%)
Baseline System	67.0	2.4	16.9	47.7	100
1. Two Stage	31.3	2.0	3.3	26.3	49.4
2. Two Stage with TINA	24.3	1.2	3.3	19.9	45.6
3. Three Stage	21.8	0.9	2.3	18.6	44.2

Table 8-4: *Breakdown of Understanding Error Rates (UER) for baseline system and three experimental systems on a 425-utterance test set with unknown city names.*

	Subs	Ins	Dels	Err
%	29.7	9.5	18.6	57.8

Table 8-5: *Error rates for Letter Recognition of Unknown Cities. There were 164 cities in total.*

sis contains an unknown word, the ANGIE parse for the word is accessed and the pre-terminal letter-phonemes are retrieved. These underlying spelling hypotheses are deduced.

In this experiment, the two-stage system attains a WER of 16.5% and a UER of 32.5%. This is a slight degradation compared with the original System 1 (15.6 WER and 31.3 UER) that uses a phoneme-based grammar in stage two. We only examine the spelling hypotheses for the subset of sentences that are correctly recognized. This ensures that the unknown words are detected in the correct location and the surrounding hypotheses are also accurate. Our evaluation is focused on the system's ability to propose correct spellings in spite of the chance that the underlying phones are incorrectly recognized. The letter error rate is then computed on the unknown words of 164 sentences. The result is a 57.8% error rate. The composition is given in Table 8-5.

Although these results remain preliminary, they nonetheless demonstrate that spelling extraction of unknown words is feasible at recognition time. Furthermore, these results were ascertained without any optimization on the sound-to-letter capabilities. Table 8-6 tabulates the top 20 letter confusions among the individual letter errors.

Number of Confusions	Confusions	Number of Confusions	Confusions
13	E ⇒ A	5	K ⇒ C
13	A ⇒ O	5	N ⇒ R
11	O ⇒ A	4	P ⇒ T
8	A ⇒ I	4	P ⇒ F
8	U ⇒ O	4	C ⇒ O
7	S ⇒ C	4	O ⇒ E
7	E ⇒ I	4	E ⇒ U
6	O ⇒ U	4	S ⇒ T
5	I ⇒ E	4	M ⇒ N
5	I ⇒ A	4	K ⇒ T

Table 8-6: *Top 20 Confusions for the Letters Hypotheses.*

Let us explore further by analyzing the characteristics of the errors made for this test set. In Table 8-7, some examples of the unknown word hypotheses are given. We did not examine the actual phonetic hypotheses nor attempt to quantify a phonetic recognition accuracy. However, by inspection, many of the spelling hypotheses suggest that the underlying phonetic sequences appear to be correctly or almost correctly proposed. This seems to be true in spite of some errors in the letter spellings. Some notable examples from Table 8-7 are *Madagascar*, *Mapleton*, *Montebello*, *Montclair*, *Qatar*, *San Ramone* and *Sedona*. We are particularly encouraged by this, as even though the quantitative letter error rate is rather high, the unknown word hypotheses may still be induced from the errorful machine hypotheses. This can perhaps be performed by matching a hypothesis with a long list of city names or alternatively one can enlist a human operator to guess the city name by examining the proposed letter sequence. Or the hypothesis can simply be used to shortlist some possible unknown city names given the state. Many possibilities exist in a real world application, where city names that are not previously encountered are expected.

8.3.5 Running Times

In further evaluations, we assess the computational requirements of the system. Given the 425-utterance test set, the average computation time per utterance is measured on a single 500MHz Pentium III machine. The computation time of each stage is measured separately. On average, there are 7.9 words in every sentence in the test set. Results are given in Table 8-8.

It should be noted that, although the baseline is equivalent to the state-of-the-art JUPITER system, real-time operation requires more aggressive pruning, whereas our baseline employs minimal pruning. Evidently, the three-stage system in total operates at a speed similar to that of the baseline. The FST-based first stage yields only 4.63 seconds per utterance, which is almost twice as fast as the time for the single-stage baseline word recognizer at 8.1 seconds per utterance. In the second stage, the ANGIE only system yields 3.1 seconds per utterance, whereas adding TINA into the search increases running time significantly. But in the third stage, running time for a TINA-based search is virtually negligible. This is attributed to the compact word graphs computed from only the top 20-best sentences in stage two.

Reference	⇒	Hypotheses
Alameda	⇒	alumida
Buford	⇒	befurt
Franklin	⇒	frankfor
Gettysburg	⇒	catlsburgh
Hanover	⇒	anover
Hatteras	⇒	sateras
Hillsboro	⇒	hillsburow
Homestead	⇒	hontstead
Huron	⇒	juran
Madagascar	⇒	madigasgar
Mapleton	⇒	mapelton
Montebello	⇒	montabellow
Montclair	⇒	monclar
Mountainview	⇒	mountonvue
Napa	⇒	napid
Parsippany	⇒	parcetone
Qatar	⇒	katar
Rancho bernardo	⇒	raktulburghargo
San ramone	⇒	sanromon
Sedona	⇒	sydona
Texaco	⇒	texico
Youngstown	⇒	janston

Table 8-7: *Examples of Letter-Spelling Hypotheses from the Two-Stage Recognizer. Spellings were extracted from the letter-phonemes at the pre-terminal layer of the* ANGIE *parse tree.*

8.4 Final Remarks

In this chapter, we have assembled a three-stage system that draws upon the developments that have been charted throughout this thesis. We sought to initially establish the validity of our notions regarding linguistic modeling and our set of novel design solutions via an experiment on recognizing and understanding JUPITER sentences with unknown city names.

As we have hoped, our three-stage system has exhibited an ability to recognize the presence of an unknown word, without adversely affecting the recognition of other parts of the sentence. Moreover, we believe that eliciting the spelling of an unknown word instantaneously is a real possibility. It is also important to emphasize that we have accomplished this with a very efficient system that operates no slower than our baseline word recognizer. The next chapter will recapitulate the contributions of this thesis and ponder on directions for the future.

System	Seconds per Utterance
Single-Stage Baseline	8.1
First Stage Only	4.63
Second Stage with ANGIE Only	3.1
Second Stage with ANGIE-TINA Only	5.0
Third stage with TINA Only	n/a

Table 8-8: *Average Computation Time Per Utterance. The test set averages 7.9 words in length.*

Chapter 9

Conclusions

9.1 Summary of Contributions

In this thesis, we have addressed how one could implement a multiple domain system with a flexible and dynamic vocabulary. Our foremost vision has been one of a system that allows a user to switch among multiple topic domains automatically and seamlessly within a single conversation. In addition, the system learns any new words encountered at the spoken input. When the system is presented with a query regarding an unknown place or person, it identifies the presence and exact location of the unknown word within the sentence. At the same time, the system proposes a phonetic baseform and orthography for the new word. Once these hypotheses are verified with the user, the system can automatically add them into the lexicon. The new word is instantaneously available to the system for future recognition.

A major challenge in implementing this vision is in tackling one of the most elusive problems present in speech understanding systems today: the OOV problem. The presence of unknown words tends to multiply errors within the in-vocabulary regions of the sentence. This may lead to a compounding of errors in the continuing dialog, often inciting user frustration. Thus, it is most important to handle OOV words in an intelligent manner. In our research, we focus on exploring the application and organization of various linguistic knowledge sources. These need to reckon with two conflicting demands: to increase flexibility in order to support sequences associated with unknown words, and to maximize constraints in order to preserve high performance on in-domain, in-vocabulary data.

In this thesis, we have combined a composite of novel ideas for combining disparate linguistic constraints. These are:

- Developing a multi-stage architecture in which the core recognition engine at the first stage focuses on exploiting low-level linguistic knowledge.
- Folding a complex hierarchical sublexical model into a novel FST representation for integration with the recognition search algorithm, thereby enhancing a low-level first-stage recognizer.
- Improving the low-level first stage with novel context-dependent linguistics units that dually encode spelling and pronunciation. These are utilized both at the phonemic level with letter-phonemes and syllabic level with novel morph units.

We will summarize these main points below.

9.1.1 A Multi-Stage Approach

We have conceived of a multi-stage architecture in which information from the linguistic hierarchy is applied successively from the bottom up, proceeding from one stage to the next. At each stage, the interface is a network or lattice. In an initial recognition engine, we utilize only low-level linguistic models to capture generic English domain-independent knowledge. The models draw upon general acoustic and linguistic knowledge, codifying phonotactic, phonological, syllable and morphological information. This information is general enough to support both out-of-vocabulary and in-vocabulary words. The first stage is envisioned to narrow the search space, outputting a subword network. The network serves to steer a second stage which consists of several parallel domain-specific recognizers. Each of these utilize higher order language models tailored exclusively to each individual topic domain.

Throughout our thesis, we have experimented with several multi-stage systems. In Chapter 4, we conducted an initial experiment where the first stage of a two-stage system utilized only syllable-level information. Meanwhile a second-stage integrated models from the subword level to the natural language level. This two-stage system demonstrated the feasibility of a multi-stage architecture. In particular, it was shown that a phonetic network served as an effective interface between the first and second stage. The phonetic network is a compact way of representing hypotheses favored by the first-stage models but does not force later stages to commit to any selections of the earlier stage, avoiding irrecoverable errors incurred by making hard decisions. Furthermore, in Chapter 5, when we enhanced the low-level first stage with hierarchical sublexical models, more evidence pointed to the viability of the two-stage paradigm. In fact, adding such models in the first stage boosted gains such that the overall performance of the two-stage system significantly outperformed that of a single-stage baseline.

9.1.2 Sublexical Modeling and Finite-State Transducers

For a number of years, we have envisioned a system which can predict structures of unknown words using a hierarchical sublexical framework known as ANGIE. ANGIE has been designed to characterize phenomena such as phonology, syllabification and morphology via a trainable context-free grammar. In the past, it was used in a number of speech applications. Here, we have enlisted it to predict phonetic sequences of an unknown word by generalizing the knowledge it has learned from in-vocabulary training data.

Our contribution has been to devise an effective architecture in which these rich hierarchical models can best be exploited to enable the recognition of unknown words within a sentence. The computational demands of the ANGIE parsing mechanism have been a stumbling block towards full integration with a traditional recognition system. In our design, the dynamic parser is integrated in a second stage where the search space has been significantly pruned down. This reduction in space renders the second stage computation much more tractable, and additionally, allows an integrated search that fully couples ANGIE parsing with the application of word-level models.

Another breakthrough has been the transformation of ANGIE models into a flattened FST structure. This was conceived with the notion that the context-rich and low-level models of ANGIE can benefit the performance of the first-stage recognizer, in affording tighter constraints. FSTs constitute a versatile and parsimonious representation for expressing ANGIE constraints, enabling their integration with other more conventional language constraints. During our work, it became evident that the task of translating the powerful ANGIE models to an FST representation would pose many challenging questions. We determined that the FST structure would need to encapsulate the vast ANGIE probability space in an efficient manner without foregoing ANGIE's ability to generalize knowledge towards unobserved phonetic sequences. This left us with the task for configuring an underlying FST structure that is compact yet replicates ANGIE's probability modeling. In Chapter 5, we first adopted an FST-based stage one where an initial attempt was made to incorporate ANGIE probabilities into the FST weights. A right-to-left branching FST captured all pronunciation variants found in training data, and pre-computed ANGIE probability scores were assigned on the FST arcs. In experiments with in-vocabulary test data, results revealed that both recognition and understanding accuracy were improved by enhancing the first stage with ANGIE probabilities, compared with a baseline that was devoid of any statistical pronunciation modeling. This indicated that the ANGIE probabilities can be an effective pronunciation model, producing additional gains over and above conventional n-gram models in a system.

However, we determined that our initial ANGIE-FST effort fell short of satisfying the criterion to support previously unobserved phonetic sequences. But in Chapter 6, we developed the innovative approach of a column bigram FST in which the ANGIE parse tree is viewed in terms of generating a sequence of vertical columns. This new approach allowed us to construct a bigram FST where the FST arcs represent transitions from one ANGIE column to another. The resultant FST accepts phonetic strings and emits pre-terminal phoneme or letter-phoneme strings as well as additional information extracted from the parse tree or columns. This method fundamentally differs from our previous in that paths in the FST are no longer strictly confined to the lexicon of the training data. In the same way that ANGIE parse trees trained on in-vocabulary data can be generated for novel phonetic sequences, novel sequences can be admitted in the column bigram FST, generating corresponding novel phoneme or letter phoneme sequences along with other parse information. Eventually, this became our method of choice for converting ANGIE into an FST structure, and we proceeded to use these FSTs to automatically generate a new set of subword lexical units.

9.1.3 Novel Symbolic Representations involving Graphemes

Another important contribution of our work has been the development of novel symbol representations that account for spelling as well as pronunciation. This involved experimenting with units at the phoneme level, as an intermediate representation embedded in the parse tree, as well as at the lexical level for our first-stage recognizer. Our initial motivation was to incorporate spelling information into the first stage, as it has the potential to enhance constraints as a source of low-level linguistic knowledge. Secondly, we were inspired to introduce grapheme information into the recognizer so that the spelling of a detected unknown word could be inferred instantaneously upon recognition. The following revisits some of these novel units of representation adopted in our work.

Letter-Phonemes

A major contribution has been the development of letter-phonemes, which amalgamate phonemes and graphemes into a single mode of representation. A single letter-phoneme captures the underlying phoneme and grapheme along with other contextual factors such as lexical stress and syllable position. A direct consequence has been the enhancement of the ANGIE models so that grapheme information is melded together with phonological and other subword constraints. Our experiments showed a drop in perplexity, suggesting a potential benefit in their adoption. Moreover, when an unknown word is detected, the letter-phoneme sequence can be extracted from the parse tree, and, immediately, a spelling hypothesis is generated. Our strategy turned out to be a particularly convenient way of representing in-vocabulary items by making the spelling accessible at the baseform. And the letter-phonemes later played a critical role in automatically generating new morphs with novel spellings.

The Morph Unit

The morph unit was conceived as a linguistically motivated syllable-like unit that embeds not only syllable information but also spelling and other contextual properties. Some of these are lexical stress information and the underlying morphological unit such as prefix and suffix. Like the syllable, the morph unit is more general than the word, and therefore more likely to support an OOV item. Yet, with the additional context, a morph supplies significantly more constraint than the syllable, and is more likely to ensure the correct recognition of in-vocabulary items. We have used morph units in the first pass throughout our various multi-stage systems. Eventually, our morphs were automatically generated as described in Chapter 7, and in our final systems described in Chapters 6 and 8, the stressed roots were decomposed into their constituent onsets and rhymes. In other words, we made use of an inventory of sub-morph and morph units in stage one. This was our solution for achieving a compromise between optimizing generality and constraint while making the resultant FSTs more compact.

Novel Subword Units

One key feature in our architecture has been a first stage that outputs a phonetic network. This meant that the first-stage subword lexicon is transparent to that of later stages, and our primary focus should center on the quality of the phonetic network. Hence, we set out to re-optimize our morph lexicon in the first stage in order to simultaneously improve the probability models and reduce the size of the ANGIE-FST.

To achieve this, we devised a procedure for implementing this generative lexicon, by building on the technology developed during the earlier part of the thesis. More specifically, an iterative algorithm was designed to discover novel subword units by processing the letter-phoneme outputs of the column bigram ANGIE-FST. After a small number of iterations, convergence was achieved. The result was a decrease in perplexity and overall reduction in the size of our FSTs.

The more intriguing outcome of this work was the nature of the novel morph units themselves. These lexical units were modified in terms of spelling and placement of syllable boundaries. Each unit contained at most a single stressed syllable. At the sentential level, a sequence of these then seemed to characterize the alternating stress properties in the sentence realization. In capturing the rhythmic patterns of the sentence, this was reminiscent of rhymic foot units described in metrical phonology. It was of interest to us that such units could be automatically discovered and possibly beneficial for recognition.

9.1.4 Demonstrating Flexible Vocabulary

This research culminated towards a final set of experiments on some test data containing unknown words. The above ideas were assembled together to build a final three-stage system. Here, the first stage utilizes the column bigram FST along with a letter-phoneme ANGIE grammar. The lexical units are precisely those derived from the automatic generation algorithm. A second stage searches through a phonetic network, applying the ANGIE parse mechanism. This stage determines possible locations of unknown words. The hypotheses are output in the form of a word graph, which is processed by a third stage where natural language information is utilized.

In experiments undertaken with JUPITER sentences containing unknown city names, we found significant reductions in understanding and recognition error when compared with a baseline that lacked any OOV handling capability. We achieved success in improving recognition of the in-vocabulary regions surrounding the unknown words, and detecting the unknown words themselves. Our ability to process a query regarding an unknown city was greatly increased. In another pilot experiment, we also demonstrated that instantaneous spelling extraction of the unknown word was a possibility.

9.2 Future Directions

The work described in this thesis has only touched upon some of the issues we are attempting to address. The results we attained have been encouraging and have opened up many possibilities for further experimentation, particularly in applying the system to a real-world speech application where flexible vocabulary is a critical issue due to the prevalence of unknown words, and automatic domain switching becomes a practical necessity.

Let us consider the results gleaned from this research and ponder the next steps one could take towards realizing a truly multi-domain flexible vocabulary system. The following will provide a flavor for some direct extensions or applications to this thesis, and give a number of suggestions for further experiments in the immediate future.

9.2.1 A Multi-Domain System

We believe that the current system is only a small step from a truly multi-domain system in terms of implementation. In order to advance further towards a multi-domain system, we would like to assemble a second stage consisting of several individual recognizers in parallel, each with its own domain-specific models. We feel that this is feasible because of the small amount of computation required in the second-stage search. Further research can focus on how to combine the outputs of each of these recognizers to reliably determine the spoken input utterance and the domain of the query. This allows for switching automatically to a different topic during the dialog. Experiments will need to study ways to combine scores output from each recognizer, and devising an algorithm that determines strategic points during the dialog for permitting domain switching while preventing spurious switching at inappropriate times.

To optimize the performance of such a system, the first stage needs to fulfill some of the requirements identified early on in this thesis. That is, the linguistic knowledge sources need to be sufficiently constraining to ensure high recognition accuracy, yet they need to be adequately flexible and general in order to cover all the topic domains that are served, as well as to cater to possible unknown words, which will be more probable and varied once multiple topic domains are possible. Hence, one could train the first stage on a larger, more general corpus. For instance, one could combine the data from all the topic domains involved, and generate using the techniques described here a new subword lexicon for the first stage. It remains to be answered whether a real domain-independent first stage which may require up to 10,000 morph or syllable units, can maintain a comparable level of performance. Experiments in this area can help determine the extent to which domain independence is possible, in a competitive system.

9.2.2 Flexible Vocabulary

Meanwhile the lessons that we have learned from this thesis can be applied to many different immediate applications where we expect to encounter unknown words in many types of queries. For instance, this can range from new users who are enrolling their names into a system to queries associated with newly emerged or unknown place names such as restaurants. One can imagine that these are frequent scenarios in domains such as travel planning, directory assistance or city guides.

Then how can we realize the automatic acquisition of new words in a real state-of-the-art system? Following are some suggestions for future work that may allow our system to be incorporated, producing real gains within the foreseeable future.

Combining with Confidence Scoring or a Rejection Mechanism

In an alternative strategy, the highest performance on in-domain utterances can be ensured only by exclusively processing sentences which have been rejected from any state-of-the-art system. In some applications, the priority is to process the in-domain queries correctly when very few OOV sentences are expected. Many systems do currently employ confidence scoring methods to reject sentences. These act to screen out sentences for which the recognizer is uncertain for various reasons. A subset of these sentences may be queries which could be salvaged by our system by accurately pinpointing the location of the unknowns, and subsequently learning their phonetic and orthographic transcriptions.

Learning an Unknown Word

Our research has only begun to address the concept of instantaneously learning the unknown word upon detection. We conducted a single experiment in which we extracted the highest scoring letter spelling hypothesis, from the proposed phonetic sequence of the unknown word. Consequently, the spellings proposed are errorful, due in part to the errorful phonetic hypotheses. As yet, our ANGIE grammar has not been formally evaluated for its sound-to-letter accuracies. In future, we hope to experiment further in processing N-best hypotheses of both letter sequences and phonetic sequences or both, in the unknown word location. It may be possible to integrate with a second ANGIE-based sound-to-letter module based on the same inventory of letter-phonemes. This can be done by an ANGIE grammar with letter-phonemes at the pre-terminals and a grapheme set as the terminal units. This grammar can be trained to generate alternative confusable spellings with probabilities given a possibly error-ridden letter-phoneme sequence.

What remains is still a reliable method to ascertain the correct spelling for the new word with certainty, which then allows subsequent incorporation into the recognizer dictionary. In the absence of a keyboard, the user cannot enter this new word herself, and the process has to be completed within the dialog. First of all, we can process the N-best list of letter spellings by comparing with a long list of possible unknown words. For instance, imagine an application where a large list of place names is easily accessible, perhaps on-line, although their baseforms are not available to the recognizer[1]. One can compute using a distance metric the best matching name between the long list and the N-best output. A more practical and robust strategy may be to find a small list of matching names and to present them to the user. With a displayful mode, the user can select the correct name from a long list. More problematic is a displayless application such as in telephony. A workable solution is then for the system to ask a user to key in the unknown name on the keypad. This will constrain the number of spelling hypotheses significantly. Upon acquisition of the new word, the various models need to be updated so that the new word can be recognized the next time it is used. More investigation is required to establish if recognition of a newly acquired word can operate effectively in an application.

[1] In most applications, it is certainly impossible to regularly update the recognizer dictionary to match items in constantly changing on-line databases.

9.2.3 Continuing Research in ANGIE Integration

Early in this thesis, we introduced the issue of incorporating ANGIE into a recognition system. Conventional recognizers do not employ any powerful low-level constraints resembling those offered by ANGIE, but instead they rely heavily on higher-level language models that are inextricably tied to the topic domain. While we believe ANGIE models can improve recognition performance, the most challenging problem has been handling the added computation that ANGIE imposes. Throughout our work, we have demonstrated that ANGIE constraints improve accuracy, and ANGIE can be substantially compacted into an FST structure. Our column bigram structure was a novel method for partially enumerating the probability space covered by ANGIE. But we pose the question of whether ANGIE can be better incorporated. Currently, in using FSTs, we are somewhat at the mercy of the optimization procedures mandated in the generic algorithms. They dictate how the probability scores are to be distributed, and how those paths are laid out. Perhaps philosophically, the ANGIE-FST lacks the elegance of the ANGIE framework. Our current FSTs instantiate the entire possible search space prior to recognition time, which is diametrically opposed to the sharing that occurs in the dependencies of the probability model in the original ANGIE mechanism.

In the future, we can keep pursuing the route of integrating the ANGIE hierarchical model into the initial stage, given that computational resources will continue to grow more abundant. In recent work, Mou [74] has suggested a unified environment for generally applying hierarchical linguistic mdoels, employing an FST framework. Using the full coverage of a hierarchical model should produce better performance, particularly in the face of unknown words. But for now, it may be wise to ponder on alternative methods to represent ANGIE efficiently in the recognizer.

Appendix A

Glossary

The following is a list of terms that have been used in this thesis. We have included this to aid the reader with some of the terminology where definitions may be ambiguous. Although this is by no means a complete list of terminology relevant to our work, we hope this short glossary will serve to clarify difficulties the reader may encounter. Many of the definitions are quoted directly from [16] and [65].

- **Ambisyllabicity**

  A principle in metrical phonology which allows intervocalic consonants to be members of both adjacent syllables in the underlying syllabification of a language, conforming to the language's syllable structure template. See [40].

- **Coda**

  A term used in phonetics and phonology to refer to the element of a syllable which follows the syllable nucleus.

- **Constraint**

  In Artificial Intelligence, a constraint is a restriction on the search space.

- **Context-free Grammar**

  A grammar in which all the rules apply regardless of context, i.e. they would be all of the type, "Rewrite X as Y", no further conditions being specified.

- **Deletion**

  When a phoneme has been deleted in its phonetic realization, little or no evidence of the phoneme can been observed in the speech signal. For example, by a rule of generative phonology, the phoneme /*t*/ might be optionally deleted from the phonological representation of the word *interested* in the following phonetic representation: /ih n rx eh s tcl t ax dcl/.

- **Demisyllable**

  This refers to units smaller than the syllable. It can refer to units ranging from the size of a phone to a constituent of the syllable. These are sometimes employed as units for modeling in engineering approaches in speech recognizers, and have not been formally defined in linguistics [84].

- **Diacritic**

  In phonetics, a mark added to a symbol to alter its value. In this thesis, diacritics are used to augment units with contextual properties. For example, the letter phoneme */b!/* is the */b/* phoneme denoted with *"!"* for syllable onset position.

- **Finite-State Automaton / Machine**

  A finite-state automaton (FSA) or a finite-state machine (FSM) can be seen as a directed graph with labels on each arc. Mathematically, (from Roche and Schabes [90]), an FSA is a 5-tuple (Σ, Q, i, F, E) where Σ is a finite set called the alphabet, Q is a finite set of states, $i \in Q$ is the initial state, $F \subseteq Q$ is the set of final states and $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the set of edges.

- **Finite-State Grammar**

  A grammar in which sentences can be characterized in terms of the transitions of an automaton from one state to another, known as a finite-state machine. Finite-state grammars are also known as regular grammars.

- **Finite-State Transducer**

  A finite-state transducer (FST) can be seen as a finite-state automaton, in which each arc is labeled by a pair of symbols rather than a single symbol. Mathematically, (from Roche and Schabes [90]), an FST is a 6-tuple $(\Sigma_1, \Sigma_2, Q, i, F, E)$ such that:

  - Σ_1 is a finite alphabet, namely the input alphabet,
  - Σ_2 is a finite alphabet, namely the output alphabet,
  - Q is a finite set of states,
  - $i \in Q$ is the initial state,
  - $F \subseteq Q$ is the set of final states,
  - $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the set of edges.

- **Foot**

A term used by some phoneticists and phonologists to describe the unit of rhythm in languages displaying isochrony, i.e. where the stressed syllables fall at approximately regular intervals throughout an utterance. The term has particular relevance in metrical phonology where it refers to an underlying unit of metrical structure (or stress-foot), consisting of syllable rhymes, and organized constituents that make up phonological words. Feet are classified as left-headed (the leftmost rhyme is stressed) or right-headed (the rightmost rhyme is stressed).

- **Grapheme**

  The minimal contrastive unit in the writing system of a language. The grapheme *a*, for example, is realized as several allographs or variants including *A* and *a*.

- **Inflectional Suffix**

  Inflectional suffixes in linguistics generally signify grammatical relationships such as plural, past tense and possession, and do not change the grammatical class of the stems to which they are attached. In our ANGIE grammar, we have defined an inflectional suffix set more generally to include the plural (*=s*), the past tense (*=ed*) as well as other common endings such as *=ful*, *=able*, *=ness* and *=ing*.

- **Isochrony**

  A term used in phonetics and phonology to refer to the rhythmic characteristics of some languages. In isochronous rhythm, the stressed syllables fall at approximately regular intervals throughout an utterance. An implication of this is that the theory predicts that unstressed syllables between stresses will be uttered in similar periods of time. If there are several unstressed syllables, accordingly they will be articulated rapidly, to get them into the time span available. Isochrony is said to be a strong tendency in English which is accordingly referred to as a stressed-timed language. The units of rhythm in such languages, i.e. the distance between stressed syllables, are called feet by some phoneticians.

- **Lax**

  One of the features of sound set up by Jakobson and Halle [43] in their distinctive feature theory of phonology, to handle manner of articulation. Lax sounds are those produced with less muscular effort and movement, and which are relatively short and indistinct, compared with tense sounds. Examples are vowels articulated nearer the center of the vowel area (as in the phoneme */ih/* in *bit* and the phoneme */uh/* in *put*.)

- **Metrical Phonology**

  A theory of phonology in which phonological strings are represented in a hierarchical manner, using such notions as segment, syllable, foot and word. Originally introduced as a hierarchical theory of stress, the approach now covers the whole domain of syllable structure and phonological boundaries. The underlying metrical structure of words and phrases may be represented in the form of a metrical tree, whose nodes reflect the relative metrical strength between sister constituents. These are described in [40, 31].

- **Morph**

  While morpheme units are abstract (see next entry), in linguistics, morphs are the real forms that represent morphemes. For example, in the word *unhelpful*, */ah n/* realizes the negative morpheme, */h eh l p/* realizes the root and */f uh l/* realizes the adjective. Morphs have been adopted in this thesis as lexical units in the recognition engine. While they loosely correspond with the linguistic definition, they are more precisely seen as syllable-sized units encoding spelling, position within word (rather than a precise morphemic function) and stress. Each morph corresponds with one or more phonemic pronuncations. One example is the word *directions* which consists of four morphs: a prefix *di-* with phonemic representation */d ay/*, a stressed root *rec+* with phonemic representation */r eh k/*, a suffix *-tion* with phonemic representation */sh! en/* and the inflectional suffix *s* with phonemic representation */z/*. The suffix *-tion* can also have an alternate pronunciation in */ch! en/* as in the word *question*.

- **Morpheme**

  The minimal distinctive unit of grammar, and the central unit of morphology. The morpheme is seen as the smallest functioning unit in the composition of units. This concept is used to interrelate notions such as root, prefix, etc. In the English word *unhelpful*, three components of meaning are present: negative plus help and adjective.

- **Morphology**

  The branch of grammar which studies the structure or forms of words, primarily through the use of the morpheme construct. The study of the meaningful parts of the words.

- **Nucleus**

  This refers to the central element of a syllable, usually consisting of a vowel or a diphthong.

- **Onset**

  This refers to the part of a syllable that precedes the vowel nucleus, e.g., the phoneme */k/* in */k ae t/* for the word *cat.*

- **Parse Tree**

  In traditional grammar, parsing refers to the pedagogical exercise of labelling the grammatical elements of a sentence, e.g. subject, predicate, past tense etc. The term parse identifies the breakdown of a text in terms of syntactic, semantic and referential information, as presented in the form of a parse tree. In this thesis, we often refer to an ANGIE parse tree which labels constituents smaller than a word, from the phonetic level upwards. We also refer to a column of a parse tree which is defined as the collection of nodes along the path from the root to a leaf or terminal node, arranged into a vertical column.

- **Phone**

  A term used in phonetics to refer to the smallest perceptible discrete segment of sound in a stream of speech. From the viewpoint of segmental phonology, phones are the physical realization of phonemes. Phonic varieties of a phoneme are referred to as allophones. In our speech recognizer, the phonetic inventory directly corresponds with the acoustic models used. Phone units are also represented in the terminal layer of the ANGIE parse tree.

- **Phoneme**

  The minimal unit in the sound system of a language, according to traditional phonological theories. Phoneme units are represented in the pre-terminal layer of the ANGIE parse tree, although we prefer to refer to the representation as *pseudo-phonemes* as they have been augmented with other contextual markers. For example the phoneme */b/* in the onset position appears as */b!/.*

- **Phonetic Network / Phonetic Lattice / Phone Graph**

  This refers to a lattice, or directed graph, where each edge or arc of the network represents a phonetic event. An edge runs between two nodes which mark the begin and end points of a segment. When used in a speech recognizer, the phonetic network restricts the entire search space to a portion that the recognizer search can visit. The network thereby steers the search towards only those hypotheses that are embedded in the network. Generally scores are also stored on the edges corresponding to the probability of the phonetic event at that segment.

- **Phonetics**

  The study of the nature, production, and perception of sounds of speech, in abstraction from the phonology of any specific language.

- **Phonology**

  A branch of linguistics which studies the sound systems of languages. The aim of phonology is to demonstrate the patterns of distinctive sound found in a language, and to make as general statements as possible about the nature of sound systems in the languages of the world.

- **Phonotactics**

  A term used in phonology to refer to the sequential arrangements of phonological units which occur in a language. In English, consonant sequences such as */fs/* and */spm/* do not occur initially in a word, and there are many restrictions on the possible consonant and vowel combinations which may occur, e.g. */ng/* occurs only after some short vowels */ih eh ae/*. These sequential constraints can be stated in terms of phonotactic rules.

- **Prosody**

  A term from suprasegmental phonetics used to refer collectively to pitch, loudness, tempo, rhythm, stress and intonation.

- **Rhyme**

  In metrical phonology, a term referring to a single constituent of syllable structure comprising the nucleus and coda; sometimes it is called the core. The notion postulates a close relationship between these two elements of the syllable, as distinct from the syllable onset.

- **Rhythmic Foot**

  See the entry for **Foot**.

- **Root**

  The root is a part of the word left when all the affixes are removed. (Affixes are simply morphological elements other than the root. These include prefixes and suffixes, for instance.) It is a base form of a word which cannot be further analyzed without total loss of identity. From the semantic point of view, the root generally carries the main component of meaning in a word.

- **Stress**

  A term used in phonetics to refer to the degree of force used in producing a syllable. Stress may be correlated with observable parameters such as length, energy and so forth. It is considered a phonological feature by which a syllable is heard as more prominent than others.

- **Sub-morph**

  This term is used exclusively in this thesis to refer to constituents that are directly derived from our morph inventory. More precisely, they are the decomposition of stressed root morphs into constituent onsets and rhymes. Consequently, we refer to the onsets and rhymes collectivelly as sub-morphs. Like the morphs, they jointly encode their position in the word and spelling.

- **Sublexical / Subword Modeling**

  In [59], Lau refers to modeling the sequence of phones permitted for different sequences of words as subword lexical linguistic modeling, alternatively coined sublexical modeling. He divides up the approaches of sublexical modeling into either an explicit pronunciation graph modeling or implicit modeling of variation.

- **Syllable**

  A unit of pronunciation larger than a single sound and smaller than a word. Syllabification is the division of a word into syllables. The basic structural possibilities of a syllable in a language are described in templates. The opening segment of a syllable is called the onset. The closing segment of the syllable is the coda and the central segment of the syllable is the nucleus.

- **Word Graph / Word Lattice / Word Network**

  This refers to a lattice, or directed graph, where each edge or arc of the network represents a word hypothesis. An edge runs between two nodes, which mark the beginning and end points of a word segment. When used in a speech recognizer, the word network restricts the entire search space to a portion that the recognizer search can visit. The network thereby steers the search towards only those chosen word sequences that are embedded in the network. Generally, scores are also stored on the edges, corresponding to the probability of a word at that segment, given the surrounding allowed transitions.

Appendix B

A Guideline on Notation

In the following, we will provide a complete list of the meanings of the annotations pertaining to our morph and letter-phoneme units. This is intended to aid the reader in deciphering the contextual markers encoded in our units, given in examples throughout this thesis.

B.1 Morphs

The following table lists each type of morph and the associated annotation.

Morph Class	**Marker**	**Example**
Stressed Root	"+" suffix	*bos+*
Stressed Onset	"=" suffix	*b=*
Stressed Rhyme	"=" prefix and "+" suffix	*=os+*
Prefix	"-" suffix	*re-*
Derivational Suffix	"-" prefix	*-tion*
Inflectional Suffix	"=" prefix	*=ed*
Unstressed Root	none	*a*

B.2 Letter-Phonemes

The following table lists the meanings of each type of marker for the letter-phoneme set.

Suffix Markers	**Meaning**	**Example**
“_l+”	stressed long vowel	*/a_l+/*
“_x+”	stressed lax vowel	*/ea_x+/*
“+”	stressed vowel but not long or lax	*/a+/*
“_uns”	unstressed vowel nucleus or rhyme	*/en_uns/*
“!”	consonant in onset position	*/b!/*
“_fcn”	vowel in a function word context	*/ee_fcn/*

Other Special Symbols	**Meaning**
*/s*pl, d*ed/*	specific to the inflectional suffix
/+nt, n+t/	+ for the apostrophe punctuation
/ˆ /	Indicator of letter being consumed by the preceding morph. E.g., *o ˆ ne+* is a stressed morph where *o=* denotes the onset and *=ˆ ne+* denotes the rhyme. The vowel in the rhyme is associated with the letter that was consumed in the preceding onset.

Appendix C

Example Context-free Rules for Letter-Phonemes

Included below are excerpts of the context-free rules used in the letter-phoneme ANGIE grammar. These details are intended to facilitate the reader in gaining a better understanding of the ANGIE mechanism. The rules are divided into two groups, low level and high level. The low level rules govern the pre-terminal (letter-phoneme units) to terminal (phone units) transition, whereas the high level rules describe the derivation from the start symbol down to the phonemics or letter-phoneme layer. Conventions for the rules are as follows:

- Lines starting with a semicolon (;) are comments.
- Rules are separated by blank lines.
- The left-hand symbol (LHS) of a rule appears on its own line, prefixed by a period (.).
- Lines following the LHS are alternative right hand sides. The alternatives are separated by either new lines or double vertical bars (||s).
- Alternative symbols are enclosed in parentheses (()s).
- Optional symbols are enclosed in brackets ([]s).

C.1 Low Level Rules

```
.a+
(aa ae)

.a_x+
eh

.ai_fcn
eh

.air_uns
ehr (r rx) || (er rx)

.an_uns
(nn n nx ax) || (eh ae ax ix) (n nx ng)

.are_fcn
aa (rx r) || rx [r] || r

.aul+
(aw aa) ll [l] || (aw aa) [l] || aa

.b!
[bcl] b || bcl

.ce
(s sh)

.ck
[kcl] (k k-) || kcl

.d*ed
(dx dcl tcl tq d -n -nx) || ix  [dcl] d  || ix (dcl dx) || tcl t

.e_fcn
(iy eh)

.ea_l+
(iy ey)

.eau_uns
ow

.eh+
(eh ehr)

.eir+
ehr (rx r)

.ell_uns
(ax ll l -ll) || ax ll [l] || ax l

.er!
(r ax -rx -r) || rx [r]

.es_uns
(eh ax)

.ey_fcn
ey

.fe
f

.ge!
[dcl] jh

.h!
(hh ax y rx ow)

.i_x+
ih

.igh_l+
ay

.ille+
ih ll [l] || ih [l]

.ir+
er [r]

.j!
(hh zh y) || [dcl] jh || (sh ch jh) y

.kn!
(n -n -nx nx -nn)

.lh!
(l ll -ll -l) || ll l

.mm
m

.nd
(n nx ng -nn dcl) || n (dcl dx) || n [dcl] (d dr)

.nt
n [tcl] t || (n nn nx ng tq tcl)  || (nn n) (dx tcl tx tq)

.o_to
(ax uw ux)

.oe_l+
iy

.ol+
ow ll [l] || ow [l] || ll || aa ll [l] || aa [l]

.oo_l+
(uw ux)

.or_fcn
ow (r rx) || rx [r] || (r er)

.ou+
(aw aa)

.oul_fcn
(uh ax)

.ow_l+
ow

.pe
[pcl] (p p-) || pcl

.qu!
(-k k k- kcl) || kcl (k- k)

.re+
(ihr uh) (rx r) || er [r]

.s!
(z s sh zh -s -z)

.she
sh

.su!
[jh] (sh ch) || (-sh -z)

.te
(dx tx tq tcl)  || [tcl] (t t- tr ch)

.ti!
```

```
[jh] (sh ch) || -sh || [tcl] ch

.u
w

.u_x+
(ah uh ux)

.ul_uns
(ax ll l -ll) || ax ll [l] || ax l

.v
v

.wh!
(w hh)

.y_l+
ay

.you_fcn
y (uw ux ax) || (ch jh) y uw

.z!
(z s)
```

C.2 High Level Rules

```
.sentence
word

.word
[pre] sroot [dsuf] [isuf] || [pre] sroot uroot [dsuf] [isuf]  || fcn [isuf] || fcn [uroot]

.fcn
[fonset] fnuc [fcoda] [fsuf]
```

```
.fsuf
i_uns nt || ill_fcn || n+t

.sroot
[onset] nuc_lax+ coda || [onset] nuc+ [coda] || [onset] lnuc+ lcoda || lnuc+

.dsuf
[uonset] dnuc [ucoda] || dnuc nuc [ucoda]

.pre
[uonset] nuc [ucoda]

.uroot
[uonset] nuc

.isuf
^ly (^est ^er) || [^pl] ^ville || ^son || ^y || ^ton || ^ing ^ton || ^pl ^ton
^th (^ly ^pl) || (^th ^ly ^past ^pl ^ing ^est ^er) || [^pl] ^past (^pl ^ly)
^ing (^pl ^ly ^est) || ^er (^pl ^past)

.^ly
l! y_uns

.^er
er_uns

.^ton
t! on_uns

.^son
s! on_uns

.^y
y_uns
```

```
.^th
[e_uns] th

.^past
d*ed

.^pl
s*pl

.^ing
ing_uns

.^est
e_uns st

.^ville
v! ille_uns

.lnuc+
oh_l+ || a_l+ || ee_l+ || e_l+ || ul_l+ || u_l+ || ai_l+ || i_l+ || oo_l+
o_l+ || igh_l+ || ei_l+ || ay_l+ || ou_l+ || ea_l+ || ue_l+ || ow_l+ || is_l+
ye_l+ || ey_l+ || oa_l+ || y_l+ || ew_l+ || eigh_l+ || oe_l+ || ioux_l+

.fnuc
uh_fcn || u_fcn || a_fcn || a_ey || i_fcn || i_ay  || are_fcn || ee_fcn
you_fcn || e_fcn || e_the || oul_fcn || o_fcn || o_to || oe_fcn || or_fcn
ro_fcn || ia_fcn || ill_fcn || ai_fcn || en_fcn || ere_fcn || ey_fcn || eah_fcn
our_fcn

.fcoda
m || n || nd || s || t || n+t || d || ll || ve || f || se || nk || nt || ch
th || +ve

.nuc_lax+
e_x+ || a_x+ || o_x+ || i_x+ || ai_x+ || ea_x+ || u_x+ || ou_x+ || oo_x+ || y_x+
^o_x+ || ;ee_x+ || or_x+
.coda
m || t || b || n || ve || c || v || ti || d || ss || f || th || k || s || ng
st || nt || g || nch || nd || nk || nn || p || pp || x || l || she || sk || ck
ct || gh || sh || me || ke || dge || n s || ff || nn s || tt || bb || se || n ce
ch || gi || ne || ll || c t || p t || gg || n ch || d s || ge || l f || m p || z
mm || tch || tte || n sk || x t || the || ce || ph || tts || is || dne || n st
ces || l d || m b

.nuc+
a+ || ou+ || or+ || air+ || al+ || er+ || r+ || o+ || ar+ || all+ || ow+ || ill+
ere+ || as+ || au+ || are+ || ah+ || ore+ || eir+ || el+ || elle+ || ell+ || ir+
oi+ || oul+ || owl+ || il+ || ur+ || ear+ || ol+ || ao+ || urr+ || elh+ || our+
eur+ || ahr+ || oll+ || arr+ || ois+ || ole+ || uer+ || orr+ || yr+ || aul+
eoul+ || ille+ || re+ || eh+ || e+re+

.dnuc
en_uns || le_uns || i_uns || o_uns || e_uns || on_uns || y_uns || al_uns || a_uns
er_uns || u_uns || an_uns || ing_uns || ia_uns || in_uns || el_uns || or_uns
ar_uns || ol_uns || ai_uns || ow_uns || re_uns || ou_uns || ay_uns || oo_uns
il_uns || ell_uns || ire_uns || ea_uns || r_uns || ure_uns || ur_uns

.onset
d! || l! || b! || p! || c! || t! || d! r || v! || n! || g! || w! || qu! ^
p! l || qu! || ge! || gi! || r! || s! || m! || h! || wh! || pp! r || z! || s! t
c! l || t! r || j! || f! || k! || g! r || th! || t! w || b! l || b! r || b! u
ch! || ch! r || c! r || tch! || s! c r || f! l || kn! || s! m || f! r || g! l
s! k ^ || s! p r || s! l || g! u || zh! || s! ph || p! r || s! p || sh! || k! u
y! || s! c || o! || ph! || pp! || p! u || p! y || ph! r || s! k || s! ch || sh! r
s! n || s! t r || su! || s! w || th! r || tw! || k! r || z! u || v! l

.lcoda
n || ne || te || de || ce || t || me || m || l || ch || ke || se || ge || s
ze || th || z || le || tte || pe || n ge || st || ve || the || g || p || be
ss || nd || k || d || fe || l d || v || ti || n gi || nt || f
```

```
.nuc
i_uns || a_uns || u_uns || on_uns || e_uns || or_uns || er_uns || air_uns
an_uns || in_uns || en_uns || ar_uns || r_uns || ol_uns || y_uns || il_uns
ow_uns || au_uns || o_uns || el_uns || ey_uns || al_uns || ur_uns || es_uns
oo_uns || re_uns || ul_uns || ay_uns || ou_uns || ell_uns || le_uns || ai_uns
ew_uns || yl_uns || eigh_uns || ir_uns

.uonset
b! || c! || m! || ti! || t! || s! || r! || k! || n! || qu! || d! r || qu! ^
s! te || l! || g! u || th! || p! r || g! || ge! || p! || t! r || d! || f!
gi! || b! r || v! || w! || ch! || c! r || zh! || u! || er! || y! || j! || f! r
tch! || g! r || h! || c! l || sh! || s! t || ph! || k! l || s! l || o! || s! p
s! r || su! r || z!

.ucoda
nt || c || ve || s || d || f || l || x || ge || n || te || b || ce || t || nd
st || m || dge || ne || sh || ck || tte || c t || pe || p t || tt || th || tts
ss || ch || z || k || n ce || g || ke || p || se || me

.fonset
b! || c! || m! || d! || f! || t! || h! || w! || s! || th! || sh! || wh! || y!
```

Bibliography

[1] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms.* Addison Wesley, Reading, MA, 1974.

[2] F. Alleva and K. Lee. Automatic new word acquisition: Spelling from acoustics. In *Proc. DARPA Speech and Natural Language Workshop Oct '89*, pages 266–270, Harwichport, MA, October 1989.

[3] A. Asadi, R. Schwartz, and J. Makhoul. Automatic detection of new words in a large vocabulary continuous speech recongition system. In *Proc. ICASSP '90*, pages 125–128, Alburquerque, NM, April 1990.

[4] A. Asadi, R. Schwartz, and J. Makhoul. Automatic modeling for adding new words to a large vocabulary continuous speech recognition system. In *Proc. ICASSP '91*, pages 305–308, Toronto, Canada, May 1991.

[5] M. Boros, M. Aretoulaki, F. Gallwitz, E. Noth, and H. Niemann. Semantic processing of out-of-vocabulary words in a spoken dialogue system. In *Proc. Eurospeech '97*, pages 1887–1890, Rhodes, Greece, September 1997.

[6] J. Caminero, C. de la Torre, L. Villarrubia, C. Martin, and L. Hernandez. Online garbage modelling with discriminant analysis for utterance verification. In *Proc. ICSLP '96*, volume 4, pages 2111–2114, Philadelphia, PA, October 1996.

[7] L. Chase. Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. Eurospeech '97*, pages 815–818, Rhodes, Greece, September 1997.

[8] N. Chomsky and M. Halle. *The Sound Pattern of English.* Harper & Row, New York, NY, 1968. republished in paperback, Cambridge, MA: MIT Press, 1991.

[9] Y. Chow and R. Schwartz. The N-best algorithm: An efficient procedure for finding top n sentence hypotheses. In *Proc. DARPA Speech and Natural Language Workshop Feb '89*, pages 199–202, Philadelphia, PA, October 1989.

[10] G. Chung. Hierarchical duration modelling for a speech recognition system. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1997.

[11] Grace Chung and Stephanie Seneff. Hierarchical duration modelling for speech recognition using the ANGIE framework. In *Proc. Eurospeech '97*, pages 1475–1478, Rhodes, Greece, September 1997.

[12] K. W. Church. *Phrase-Structure Parsing: A Method for Taking Advantage of Allophonic Constraints.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, January 1983.

[13] G. N. Clements and S. J. Keyser. *CV Phonology, A Generative Theory of the Syllable.* Linguistic Inquiry, Cambridge, MA, 1983.

[14] M. H. Cohen. *Phonological Structures for Speech Recognition.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1989.

[15] R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, editors. *Survey of the State of the Art in Human Language Technology.* Cambridge University Press, Cambridge, UK, 1997.

[16] D. Crystal. *A Dictionary of Linguistics and Phonetics.* Basil Blackwell Ltd, Cambridge, MA, 1991.

[17] G. Damnati and F. Panaget. Adding new words in a spoken dialogue system vocabulary using conceptual information and derived class-based language model. In *Proc. ICSLP '96*, volume 1, pages 257–260, Philadelphia, PA, October 1996.

[18] R. De Mori and M. Galler. The use of syllable phonotactics for word hypothesization. In *Proc. ICASSP '96*, pages 877–880, Atlanta, GA, May 1996.

[19] E. C. Fudge. *Phonology: selected readings.* Penguin, Hammondsworth, 1973.

[20] O. Fujimura. Syllable as a unit of speech recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 23(1), February 1975.

[21] O. Fujimura and J. Lovins. Syllables as concatenative phonetic units. In *Syllabeles as Concatenative Phonetic Units*, pages 337–385. Foris, Dordrecht, Holland, 1978.

[22] F. Gallwitz, M. Aretoulaki, M. Boros, J. Haas, S. Harbeck, R. Huber, H. Niemann, and E. Noth. The erlangen spoken dialogue system EVAR: A state-of-the-art information retrieval system. In *Proc. ISSD '98*, pages 19–26, Sydney, Australia, November 1998.

[23] J. L. Gauvain, L. F. Lamel, G. Adda, and M. Adda-Decker. Speaker-independent continuous speech dictation. *Speech Communication*, 15:21–38, October 1994.

[24] P. Geutner. Introducing linguistic constraints into statistical language modeling. In *Proc. ICSLP '96*, volume 1, pages 402–405, Philadelphia, PA, October 1996.

[25] H. J. Giegerich. *Metrical Phonology and Phonological Structure: German and English.* Cambridge University Press, Cambridge, MA, 1985.

[26] J. Glass. Challenges for spoken dialogue systems. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop '99*, Keystone, CO, December 1999.

[27] J. Glass, G. Flammia, D. Goodine, M. Philips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue. Multilingual spoken language understanding in the MIT VOYAGER system. *Speech Communication*, 17:1–19, 1995.

[28] J. Glass and T. J. Hazen. Telephone-based conversational speech recognition in the JUPITER domain. In *Proc. ICSLP '98*, volume 4, pages 1327–1330, Sydney, Australia, December 1998.

[29] J. Glass, J. Polifroni, and S. Seneff. Multilingual language generation across multiple domains. In *Proc. ICSLP '94*, pages 983–986, Yokohama, Japan, September 1994.

[30] D. Goddeau and V. Zue. Integrating probabilistic LR parsing into speech understanding systems. In *Proc. ICASSP '92*, pages 181–184, San Francisco, CA, March 1992.

[31] J. A. Goldsmith. *Autosegmental and MMetrical Phonology.* Blackwell, Cambridge, MA, 1990.

[32] D. Goodine, S. Seneff, L. Hirschman, and M. Phillips. Full integration of speech and langauge understanding in the MIT spoken language system. In *Proc. Eurospeech '91*, Genova, Italy, sep 1991.

[33] S. Greenburg and B. E. D. Kingsbury. The modulation spectrogram: In pursuit of an invariant representation of speech. In *Proc. ICASSP '97*, pages 1647–1650, Munich, Germany, April 1997.

[34] R. Haeb-Umbach, P. Beyerlein, and E. Thelen. Automatic transcription of unknown words in a speech recognition system. In *Proc. ICASSP '95*, Detroit, MI, May 1995.

[35] M. Harper, L. H. Jamieson, C. B. Zolttowski, and R. A. Helzerman. Semantics and constraint parsing of word graphs. In *Proc. ICASSP '93*, pages I–63–I–66, Minneapolis, MN, April 1993.

[36] A. Hauenstein. Using syllables in a hybrid HMM-ANN recognition system. In *Proc. Eurospeech '97*, pages 1203–1206, Rhodes, Greece, September 1997.

[37] S. Hayamizu, K. Itou, and K. Tanaka. Detection of unknown words in large vocabulary speech recognition. In *Proc. Eurospeech '93*, pages 2113–2116, Berlin, Germany, September 1993.

[38] I. L. Hetherington. *The Problem of New, Out-of-Vocabulary Words in Spoken Language Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, October 1994.

[39] L. Hirschman, S. Seneff, D. Goodine, and M. Phillips. Integrating syntax and semantics into spoken language understanding. In *Proc. DARPA Speech and Natural Language Workshop Feb '91*, Pacific Grove, CA, February 1991.

[40] R. Hogg and C. B. McCully. *Metrical Phonology: A Course Book.* Cambridge University Press, Cambridge, MA, 1987.

[41] Z. Hu, J. Schalkwyk, E. Barnard, and R. A. Cole. Speech recognition using syllable-like units. In *Proc. ICSLP '96*, volume 2, pages 1117–1120, Philadelphia, PA, October 1996.

[42] D. P. Huttenlocher and V. Zue. A model of lexical access from partial phonetic information. In *Proc. ICASSP '84*, pages 26.4.1–4, San Diego, CA, March 1984.

[43] R. Jakobson and M. Halle. *Fundamentals of Language.* The Hague, Mouton, 1956.

[44] F. Jelinek. Continuous speech recognition by statistical methods. *Proc. IEEE*, 64, 1976.

[45] F. Jelinek and C. Chelba. Putting language into language modeling. In *Proc. Eurospeech '97*, pages KN–1–KN–4, Budapest, Hungary, September 1999.

[46] F. Jelinek, R. Mercer, and S. Roukos. Classifying words for improved statistical language models. In *Proc. ICASSP '90*, pages 621–624, Alburquerque, NM, April 1990.

[47] M. Johnson, M. Harper, and L. Jamison. Interfacing acoustic models with natural language processing systems. In *Proc. ICSLP '98*, volume 6, pages 2419–2422, Sydney, Australia, December 1998.

[48] M. Jones and P. C. Woodland. Modelling syllable characteristics to improve a large vocabulary continuous speech recognizer. In *Proc. ICSLP '94*, pages 2171–2174, Yokohama, Japan, September 1994.

[49] R. Jones, S. Downey, and J. Mason. Continuous speech recognition using syllables. In *Proc. Eurospeech '97*, pages 1171–1174, Rhodes, Greece, September 1997.

[50] R. J. Jones. *Syllable-based word recognition.* PhD thesis, Department of Electrical and Electronic Engineering, Univeristy of Wales Swansea, Wales, U.K., 1996.

[51] D. Kahn. *Syllable-based Generalizations in English Phonology.* PhD thesis, Department of Linguistics and Philosophy, Massachusetts Institute of Technology, Cambridge, MA, 1976.

[52] T. Kemp and A. Jusek. Modelling unknown words in spontaneous speech. In *Proc. ICASSP '96*, Atlanta, GA, May 1996.

[53] P. Kenny, P. Labute, Z. Li, and D. O'Shaughnessy. New graph search techniques for speech recognition. In *Proc. ICASSP '94*, pages 553–556, Adelaide, Austrailia, April 1994.

[54] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan. Speech recognition via phonetically featured syllables. In *Proc. ICSLP '98*, volume 3, pages 1031–1033, Sydney, Australia, December 1998.

[55] P. Kiparsky. From cyclic phonology to lexical phonology. In *The Structure of Phonological Representations (Part I)*, pages 131–175. Foris, Dordrecht, Holland, 1982.

[56] K. Kirchoff. Syllable-level desynchronisation of phonetic features for speech recognition. In *Proc. ICSLP '96*, Philadelphia, PA, October 1996.

[57] D. Klakow, G. Rose, and X. Aubert. OOV detection in large vocabulary system using automatically defined word-fragments as fillers. In *Proc. Eurospeech '97*, pages 49–52, Budapest, Hungary, September 1999.

[58] R. Kompe. *Prosody in Speech Understanding Systems.* Springer Verlag, Berlin, Germany, 1997.

[59] R. Lau. *Subword Lexical Modelling for Speech Recognition.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1998.

[60] R. Lau and S. Seneff. Providing sublexical constraints for word spotting within the ANGIE framework. In *Proc. Eurospeech '97*, pages 263–266, Rhodes, Greece, September 1997.

[61] R. Lau and S. Seneff. A unified framework for sublexical and linguistic modeling supporting flexible vocabulary speech understanding. In *Proc. ICSLP '98*, volume 6, pages 2443–2446, Sydney, Australia, December 1998.

[62] K.F. Lee, H.W. Hon, and R. Reddy. An overview of the SPHINX speech recognition system. In *Readings in Speech Recognition*, pages 600–610. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[63] M. Liberman and A. Prince. On stress and linguistic rhythm. *Linguistic Inquiry*, 8:249–336, 1977.

[64] B. Lowerre and R. Reddy. The Harpy speech understanding system. In *Readings in Speech Recognition*, pages 576–586. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[65] P. H. Matthews. *A Concise Oxford Dictionary of Linguistics.* Oxford University Press, New York, NY, 1997.

[66] D. McAllaster, L. Gillack, F. Scattone, and M. Newman. Fabricating conversational speech data with acoustic models: A program to examine model-data mismatch. In *Proc. ICSLP '98*, volume 5, pages 1847–1850, Sydney, Australia, December 1998.

[67] J. McCarthy. On stress and syllabification. *Linguistic Inquiry*, 10:443–465, 1979.

[68] R. Meliani and D. O'Shaughnessy. New efficient fillers for unlimited word recognition. In *Proc. ICSLP '96*, Philadelphia, PA, October 1996.

[69] H. Meng. *Phonological Parsing for Bi-directional Letter-to-Sound / Sound-to-Letter Generation.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1995.

[70] H. M. Meng, S. Hunnicutt, S. Seneff, and V. Zue. Reversible letter-to-sound / sound-to-letter generation based on parsing word morpology. *Speech Communication*, 18:47–63, 1996.

[71] Mehryar Mohri and Michael Riley. Weighted determinization and minimization for large vocabulary speech recognition. In *Proc. Eurospeech '97*, pages 131–134, Rhodes, Greece, September 1997.

[72] R. Moore, D. Appelt, J. Dowding, J. Gawron, and D. Moran. Combining linguistic and statistical knowledge sources in natural language processing for ATIS. In *Proc. ARPA Spoken Language Systems Technology Workshop '95*, pages 261–264, Austin, TX, Jan 1995.

[73] R. Moore, M. Cohen, V. Abrash, D. Appelt, H. Bratt, J. Butzberger, L. Cherny, J. Dowding, H. Franco, J. Gawron, and D. Moran. SRI's recent progress on the ATIS task. In *Proc. Spoken Language Systems Technology Workshop '94*, pages 72–75, Plainsboro, NJ, Mar 1994.

[74] X. Mou, S. Seneff, and V. Zue. Context-dependent probabilistic hierarchical sub-lexical modelling using finite-state transducers. In *Proc. ICASSP '01*, Salt Lake City, UT, June 2001.

[75] M. Oerder and H. Ney. Word graphs: An efficient interface between continuous speech recognition and language understanding. In *Proc. ICASSP '93*, pages II–119–II–122, Minneapolis, MN, April 1993.

[76] S. Ortmanns and H. Ney. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, pages 43–72, 1997.

[77] B. Oshika, V. Zue, R. Weeks, H. Nue, and J. Auerbach. The role of phonological rules in speech understanding research. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-23(1):104–112, August 1975.

[78] M. Ostendorf. Moving beyond the beads on a string model of speech. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop '99*, pages 79–83, Keystone, CO, December 1999.

[79] A. D. Parmar. A semi-automatic system for the syllabification and stress assignment of large lexicons. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1997.

[80] D. B. Paul. Algorithms for an optimal A^* search and linearizing the search in the stack decoder. In *Proc. ICASSP '91*, pages 693–696, Toronto, Canada, May 1991.

[81] D. B. Paul. Efficient A^* stack decoder algorithm for continuous speech recognition with a stochastic language model. Technical Report TR 930, MIT Lincoln Laboratory, July 1991.

[82] D. B. Paul. An efficient A^* stack decoder algorithm for continuous speech recognition with a stochastic language model. In *Proc. ICASSP '92*, pages 25–28, San Francisco, CA, March 1992.

[83] F. Pereira, M. Riley, and R. Sproat. Weighted rational transductions and their application to human language processing. In *Proc. ARPA Human Language Technology Workshop '94*, pages 249–254, Princeton, NJ, March 1994.

[84] T. Pfau, M. Beham, W. Reichl, and G. Ruske. Creating large subword units for speech recognition. In *Proc. Eurospeech '97*, pages 1191–1194, Rhodes, Greece, September 1997.

[85] P. Price. Evaluation of spoken language systems: the atis domain. In *Proc. DARPA Speech and Natural Language Workshop Jun '90*, pages 91–95, Philadelphia, PA, June 1990.

[86] B. Ramabhadran and A. Ittycheriah. Phonological rules for enhancing acoustic enrollment of unknown words. In *Proc. ICSLP '98*, volume 3, pages 819–822, Sydney, Australia, December 1998.

[87] M. A. Randolph. *Syllable-based Constraints on Properties of English Sounds.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 1989.

[88] M. Riley. Stochastic pronunciation modelling from hand-labelled phonetic corpora. In *Proc. ESCA Workshop for Modelling Prunciation Variation for Automatic Speech Recognition*, pages 109–119, 1998.

[89] M. Riley, A. Ljolje, D. Hindle, and F. Pereira. The att 60,000 word speech-to-text system. In *Proc. Eurospeech '95*, pages 207–210, Madrid, Spain, September 1995.

[90] E. Roche and Y. Schabes, editors. *Finite-State Language Processing.* MIT Press, Cambridge, MA, 1997.

[91] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech recognition. In *Proc. ICASSP '97*, pages 875–878, Munich, Germany, April 1997.

[92] C. Schillo, G. Fink, and F. Kummert. Grapheme based speech recognition for large vocabularies. In *Proc. ICSLP '00*, Beijing, China, October 2000.

[93] E.O. Selkirk. The syllable. In *The Structure of Phonological Representations (Part II)*, pages 337–385. Foris, Dordrecht, Holland, 1982.

[94] S. Seneff. Robust parsing for spoken language systems. In *Proc. ICASSP '92*, pages 189–193, San Francisco, CA, March 1992.

[95] S. Seneff. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86, March 1992.

[96] S. Seneff. The use of linguistic hierarchies in speech understanding. In *Proc. ICSLP '98*, Sydney, Australia, December 1998.

[97] S. Seneff, R. Lau, and H. Meng. ANGIE: A new framework for speech analysis based on morpho-phonological modelling. In *Proc. ICSLP '96*, volume 1, pages 110–113, Philadelphia, PA, October 1996.

[98] S. Seneff, M. McCandless, and V. Zue. Integrating natural language into the word graph search for simultaneous speech recognition and understanding. In *Proc. Eurospeech '95*, pages 1781–1784, Madrid, Spain, September 1995.

[99] E. C. Shukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck. Acoustic modeling of sub-word units in the ISADORA speech recognizer. In *Proc. ICASSP '92*, pages 577–580, San Francisco, CA, March 1992.

[100] B. Suhm. Detection and transcription of new words. In *Proc. Eurospeech '93*, pages 2179–2182, Berlin, Germany, September 1993.

[101] W. Ward. Integrating semantic constraints into the SPHINX-II recognition search. In *Proc. ICASSP '94*, pages 17–20, Adelaide, Austrailia, April 1994.

[102] V. Warnke, F. Gallwitz, A. Batliner, J. Buckow, R. Huber, E. Noth, and A Hothker. Integrating multiple knowledge sources for word hypotheses graph interpretation. In *Proc. Eurospeech '97*, pages 235–238, Budapest, Hungary, September 1999.

[103] M. Weintraub and J. Bernstein. Rule: A system for constructing recognition lexicons. In *Proc. DARPA Speech Recognition Workshop Oct '87*, pages 44–48, Harwichport, MA, October 1987.

[104] S. Wu, N. Morgan B. E. D. Kingsbury, and S. Greenberg. Incorporating information from syllable-lenght time scales into automatic speech recognition. In *Proc. ICASSP '98*, pages 721–724, Munich, Germany, April 1998.

[105] S. Wu, B. Kingsbury, N. Morgan, and S. Greenberg. Performance improvements through combining phone and syllable-scale information in automatic speech recognition. In *Proc. ICSLP '98*, volume 2, pages 459–462, Sydney, Australia, December 1998.

[106] V. Zue. The use of speech knowledge in automatic speech recognition. *Proc. IEEE*, 73(11):1602–1615, November 1985.

[107] V. Zue. Conversational interfaces: Advances and challenges. In *Proc. Eurospeech '97*, Rhodes, Greece, September 1997.

[108] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Integration of speech recognition and natural language processing in the MIT VOYAGER system. In *Proc. ICASSP '91*, pages 713–716, Toronto, Canada, May 1991.

[109] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff. The SUMMIT speech recognition system: Phonological modelling and lexical access. In *Proc. ICASSP '90*, pages 49–52, Alburquerque, NM, April 1990.

[110] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen, and I. L. Hetherington. Jupiter: A telephone-based conversational interface for weather information. *IEEE Trans. Speech and Audio Processing*, 8(1):85–96, 2000.

[111] V. Zue, S. Seneff, J. Polifroni, M. Philips, C. Pao, D. Goodine, D. Goddeau, and J. Glass. Pegasus: A spoken dialogue interface for on-line air travel planning. In *Proc. International Symposium on Spoken Dialogue Systems*, Tokyo, Japan, November 1993.

[112] V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goddeau, J. Glass, and E. Brill. The MIT ATIS system: December 1993 progress report. In *Proc. ARPA Spoken Language Technology Workshop '94*, pages 66–71, Plainsboro, NJ, March 1994.