Towards a Unified Framework for Sub-lexical and Supra-lexical Linguistic Modeling

by

Xiaolong Mou

M.S., Tsinghua University, Beijing, China (1998)
B.S., Tsinghua University, Beijing, China (1996)

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 2002

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

Certified by .....
Victor Zue
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

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

Accepted by .....
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Towards a Unified Framework for Sub-lexical and Supra-lexical Linguistic Modeling

by

Xiaolong Mou

Submitted to the Department of Electrical Engineering and Computer Science
in May, 2002 in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Abstract

Conversational interfaces have received much attention as a promising natural communication channel between humans and computers. A typical conversational interface consists of three major systems: speech understanding, dialog management and spoken language generation. In such a conversational interface, speech recognition as the front-end of speech understanding remains to be one of the fundamental challenges for establishing robust and effective human/computer communications. On the one hand, the speech recognition component in a conversational interface lives in a rich system environment. Diverse sources of knowledge are available and can potentially be beneficial to its robustness and accuracy. For example, the natural language understanding component can provide linguistic knowledge in syntax and semantics that helps constrain the recognition search space. On the other hand, the speech recognition component also faces the challenge of spontaneous speech, and it is important to address the casualness of speech using the knowledge sources available. For example, sub-lexical linguistic information would be very useful in providing linguistic support for previously unseen words, and dynamic reliability modeling may help improve recognition robustness for poorly articulated speech.

In this thesis, we mainly focused on the integration of knowledge sources within the speech understanding system of a conversational interface. More specifically, we studied the formalization and integration of hierarchical linguistic knowledge at both the sub-lexical level and the supra-lexical level, and proposed a unified framework for integrating hierarchical linguistic knowledge in speech recognition using layered finite-state transducers (FSTs). Within the proposed framework, we developed context-dependent hierarchical linguistic models at both sub-lexical and supra-lexical levels. FSTs were designed and constructed to encode both structure and probability constraints provided by the hierarchical linguistic models. We also studied empirically the feasibility and effectiveness of integrating hierarchical linguistic knowledge into speech recognition using the proposed framework. We found that, at the sub-lexical level, hierarchical linguistic modeling is effective in providing generic sub-word structure and probability constraints. Since such constraints are not restricted to a fixed system vocabulary, they can help the recognizer correctly identify previously unseen words. Together with the unknown word support from natural language understanding, a conversational interface would be able to deal with unknown words better, and can possibly incorporate them into the active recognition vocabulary on-the-fly. At the

supra-lexical level, experimental results showed that the shallow parsing model built within the proposed layered FST framework with top-level n-gram probabilities and phrase-level context-dependent probabilities was able to reduce recognition errors, compared to a class n-gram model of the same order. However, we also found that its application can be limited by the complexity of the composed FSTs. This suggests that, with a much more complex grammar at the supra-lexical level, a proper tradeoff between tight knowledge integration and system complexity becomes more important.

Another important aspect of achieving more accurate and robust speech recognition is the integration of acoustic knowledge. Typically, along a recognizer's search path, some acoustic units are modeled more reliably than others, due to differences in their acoustic-phonetic features and many other factors. We presented a dynamic reliability scoring scheme which can help adjust partial path scores while the recognizer searches through the composed lexical and acoustic-phonetic network. The reliability models were trained on acoustic scores of a correct arc and its immediate competing arcs extending the current partial path. During recognition, if, according to the trained reliability models, an arc can be more easily distinguished from the competing alternatives, that arc is then more likely to be in the right path, and the partial path score can be adjusted accordingly, to reflect such acoustic model reliability information. We applied this reliability scoring scheme in two weather information domains. The first one is the JUPITER system in English, and the second one is the PANDA system in Mandarin Chinese. We demonstrated the effectiveness of the dynamic reliability modeling approach in both cases.

Thesis Supervisor: Victor Zue
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Acknowledgments

First, I would like to thank my thesis advisor Victor Zue, who offered me the opportunity of gaining deep insights and broad experiences in the research of spoken language systems. He encouraged me to choose a research direction of my interest, and continued guiding my thesis research through the years. It was his encouragement, guidance, support and patience that made my progress in this thesis possible.

I would like to express my deep gratitude to Stephanie Seneff, who co-supervised the thesis research. Stephanie has been keeping close interaction with me throughout this research. She offered valuable suggestions, which were extremely helpful and inspiring. I have been truly fortunate to be able to benefit from her extensive experiences and remarkable enthusiasm. I would also like to thank Prof. Leslie Kaelbling on my thesis committee for her feedback on my thesis research.

I would further like to thank my former master's and bachelor's degree research advisors in Tsinghua University, Beijing, China. Prof. Wenhu Wu, Fang Zheng, Ditang Fang, and Qixiu Hu, thank you for opening the door and leading me to the field of speech-based human/computer interaction.

I would like to offer my special thanks to Chao Wang, Lee Hetherington, TJ Hazen, Jim Glass, and Joe Polifroni who assisted me in understanding and using the tools and libraries that greatly facilitated this research.

I am also grateful to all the SLS group members, who provided various perspectives on my research. Thanks to Issam Bazzi, who offered many constructive suggestions. Many thanks to my past and present officemates, Grace Chung, Atiwong Suchato, Edward Filisko, Brooke Cowan, and Vladislav Gabovich (and the Rabbit Pez Popper and the Koosh), who made my daily life a great experience.

Finally, I would especially like to acknowledge the love and support from my parents, my wife Cathy, and my friends. Thank you for giving me strength during difficult times and sharing the pleasure of my progress.

This research was supported by a contract from the Industrial Technology Research Institute, and by DARPA under contract N66001-99-1-8904 monitored through Naval Command, Control and Ocean Surveillance Center.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, speech-based interfaces have become a realistic communication channel between humans and computers. Many prototype speech systems have been developed within the research community, and the efforts have generated valid technologies that are now used in commercially available products. While early speech-based interfaces mainly focus on speech recognition (speech-to-text) and speech synthesis (text-to-speech), an increasing amount of research has been conducted recently in the effort of creating *conversational* interfaces that facilitate solving problems in an interactive manner [113]. This area has received much attention with the fast growth of on-line information. In order to access vast amounts of information more accurately and effectively, automated systems are widely used to help organize, manage and retrieve data. However, the interface between users and automated agents still falls far short of the idealized usability goal. Simple interfaces such as touch-tone menus using an Interactive Voice Response (IVR) system suffer greatly from constrained functionality, while more flexible ones such as advanced Graphical User Interfaces (GUIs) are often less intuitive to use for solving complex tasks. Conversational interfaces address these problems by providing a mechanism of communicating with automated agents interactively using natural speech, which may greatly improve the accessibility of automated information systems. Many application domains have been explored by researchers, including flight information [85], weather information [116], air travel planning [100, 101], off-line delegation [96], etc. A growing industrial interest along this direction is also emerging.

A typical conversational interface consists of three major components: speech under-

standing, dialog management and spoken language generation. The speech understanding component converts spoken inputs to formal meaning representations, which are needed by computers to understand and further process users' requests. It includes a speech recognition component that proposes sentence hypotheses from input speech signals, and a natural language understanding component that provides a meaning analysis of the hypothesized sentences. In the general case, it is unlikely that a user can successfully access the desired information or complete a transaction with just a single query. Therefore, a dialog management system is necessary to control the progression of the human/computer interaction. It maintains the dialog history, interprets users' responses according to discourse contexts, and decides how the dialog can proceed. The spoken language generation component consists of a language generator and a speech synthesizer, which produce spoken responses for requested information. It also generates additional clarification queries or feedback to users as demanded by dialog control. Figure 1-1 shows a simple schema of a conversational interface.

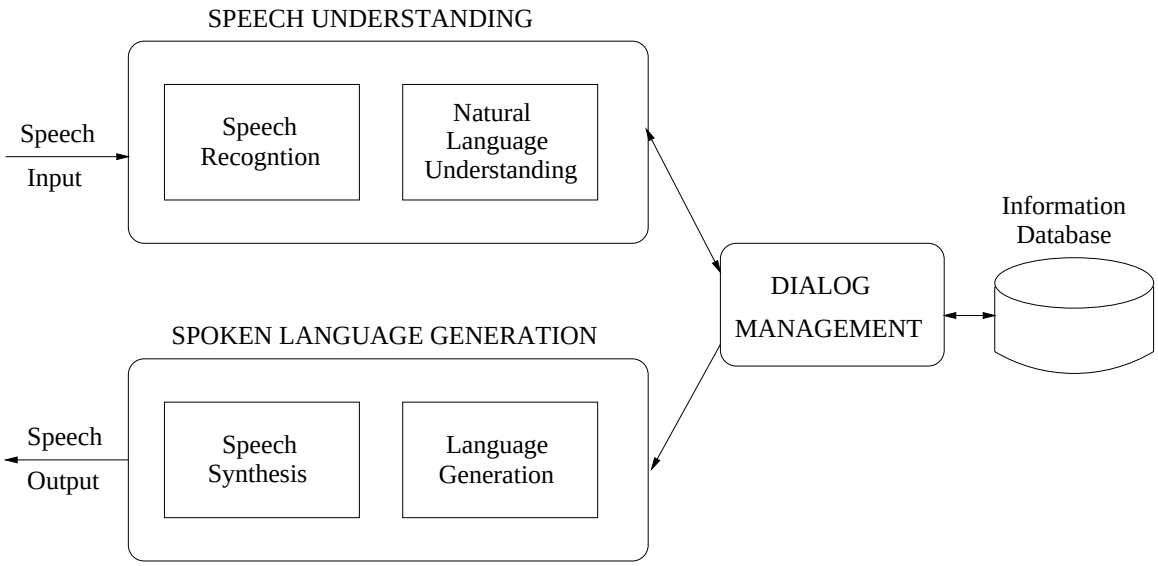


Figure 1-1: A typical conversational interface architecture.

There is no doubt that there are important and as yet unsolved problems in all aspects of conversational interfaces. However, the speech recognition component remains one of the fundamental challenges for establishing robust and effective human/computer communications. In contrast to a speech-to-text interface, the speech recognition component in a conversational interface lives in a much richer system environment. Diverse sources of knowledge are available to speech recognition and can potentially be helpful to improve its robustness and accuracy. For example, natural language understanding provides linguistic constraints in syntax and semantics, and dialog management keeps track of dialog states and discourse contexts, etc. On the other hand, however, speech recognition in conversational interfaces also faces a more challenging situation: it needs to handle spontaneous speech, i.e., speech with various co-articulation effects (such as gemination, palatalization, and nasalization in different contexts), disfluencies, false starts, previously unseen words, etc. It is therefore important to address the casualness of speech using the knowledge sources available. For example, sub-word structure information of the recognition language would be very useful in providing linguistic support for unseen words, since such generic sub-word structural constraints are not restricted to a specific recognition vocabulary. Another example is the use of confidence modeling [111, 82, 48] to improve the recognition robustness for poorly articulated speech. In this approach, the recognition result is evaluated by scoring a set of chosen confidence features against an established confidence model, and mis-recognized words can be rejected according to the confidence score. Confidence measurements can also be dynamically applied during the recognition search to help reduce recognition errors. We refer to such an approach as dynamic reliability modeling [77].

In this thesis, we mainly focus on the integration of knowledge sources within the speech understanding component of a conversational interface. More specifically, we study the application of linguistic knowledge in speech recognition at both sub-lexical (below the word) and supra-lexical (above the word) levels, and propose a unified framework for integrating hierarchical linguistic knowledge in speech recognition. We also include our research on dynamic reliability modeling that tries to apply knowledge of acoustic reliability. In the following sections, we will give an overview of a typical speech understanding system, then discuss in more detail the use of sub-lexical and supra-lexical linguistic knowledge, and how

they can be integrated in speech recognition. Next, we will briefly introduce the concept of dynamic reliability modeling. Finally, we will summarize the motivations and objectives of this research, and present a thesis organization outline.

1.1 Speech Understanding System Overview

Speech understanding is one of the major components of a conversational interface. It takes in spoken input from users, and eventually generates formal meaning representations needed by database access and dialog control. Typically, two components are involved in this process, a speech recognition component that converts the speech signal to sentence hypotheses, and a natural language understanding component that provides a meaning analysis. Figure 1-2 shows a block diagram of a typical speech understanding system.

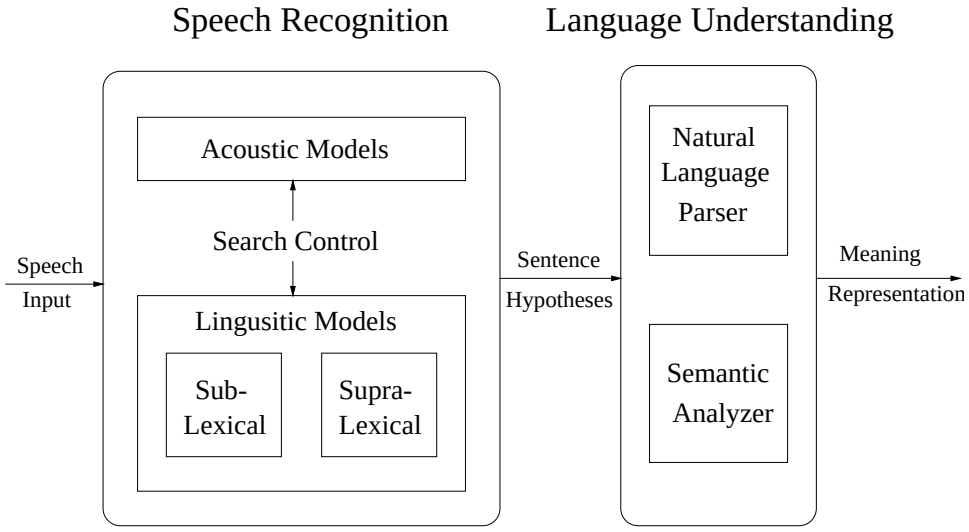


Figure 1-2: A typical speech understanding system with two major components, speech recognition and natural language understanding.

The knowledge sources applied in speech recognition can be divided into two major categories, acoustic knowledge and linguistic knowledge. They are formally represented by acoustic models and linguistic models, respectively. Acoustic models are used to characterize

acoustic features extracted from the input speech signal. They specify the likelihood that a particular linguistic event (e.g., a sequence of words or sub-word units) has generated the input sequence of acoustic features. In most of the recent systems, acoustic models are frame-based hidden Markov models (HMMs). The first applications of HMMs in speech were published by Baker [7] and Jelinek [45, 44] in the 1970s. Some other types of acoustic models have also been demonstrated to be successful, such as the feature-based segment modeling approaches [26, 115]. For a small-vocabulary system, it is possible to train acoustic models for every word. However, with the increase of vocabulary size, most modern large-vocabulary speech recognition systems build acoustic models based on sub-word units such as phones (i.e., the smallest perceptible discrete sound segments in a stream of speech). There are several advantages to using sub-word units. For example, sub-word models can be shared across different words, which helps alleviate the sparse training data problem. Furthermore, a sub-word based system has the potential of supporting previously unseen words through the concatenation of sub-word units, subject to certain constraints of the sub-word structure.

Linguistic models in speech recognition characterize the *a priori* linguistic knowledge of a recognition domain. Given the domain, they specify the likelihood that a particular linguistic event (e.g., a sequence of words or sub-word units) would occur in this domain. These *a priori* linguistic probabilities are combined with acoustic model probabilities to identify sentences that are the most probable generators of the input speech. There are two levels of linguistic models, sub-lexical and supra-lexical. The goal of sub-lexical linguistic modeling is to robustly model the construction of words from sub-word units using available linguistic knowledge. For example, under different phonetic contexts, different phone realizations are possible, such as flapping or glottal stop insertion. When acoustic models are established at the phone level, dealing with such variations is an important aspect of sub-lexical linguistic modeling. Supra-lexical linguistic models model the construction of sentences from words. The most commonly used supra-lexical linguistic model is the stochastic n-gram language model, which specifies the probability of the next word given its previous $n-1$ word context. Such a flat n-gram model does not provide support for the sentence structure, and researchers have also introduced formal grammars to address

the sentence structural constraints. More details of sub-lexical and supra-lexical linguistic modeling approaches will be presented in the next section.

The acoustic and linguistic models are combined while a speech recognizer searches through possible hypotheses and tries to find the most probable candidate. One way of integrating linguistic knowledge is to build a stochastic finite-state network that formalizes the linguistic constraints. Linguistic model probabilities are encoded as transition weights between states. Each path (i.e., a sequence of states and transitions associated with the probability weights) in the network corresponds to a possible match between the input speech and a hypothesized sentence. A match is evaluated by the path score obtained through combining the corresponding acoustic and linguistic model scores along the path. While it is relatively easy to convert some commonly used linguistic models such as n-gram into a finite-state network, it is not straightforward to incorporate more complex ones such as stochastic *hierarchical* linguistic models into such a network. These complex linguistic models are typically implemented separately, and they interact with the recognition search using customized interfaces.

Natural language understanding is responsible for generating formal meaning representations from sentence hypotheses given by speech recognizers. It typically consists of a natural language parser that performs syntactic analysis and a semantic analyzer that performs semantic analysis from the resulting parse tree. Usually the natural language parser is based on context-free grammars, which are used to describe sentence structure. While typical generic natural language processing systems are designed for well-formed written text, language processing in a speech understanding system has to deal with the spontaneity of speech and possible recognition errors of input sentences. This has led to a tendency of using semantic-driven rather than syntax-driven approaches for language processing in speech understanding systems [113]. The main idea is that, if a complete syntactic analysis is not possible, which is likely to happen in the presence of recognition errors, complete or partial meaning can still be derived from key phrases and words. Note that syntactic structures tend to be domain-independent, and capture general syntactic constraints of the language. The semantic structures tend to be domain-specific, and capture semantic constraints in a specific application domain. It is possible to combine the syntax-driven and

semantic-driven approaches, thus taking advantage of both of them. More details of natural language understanding are discussed in Section 4.3.

1.2 Current Landscape in Linguistic Modeling and Natural Language Understanding

In this section, we will summarize current efforts of linguistic modeling at the sub-lexical and supra-lexical levels, and briefly discuss the natural language understanding approaches used in a speech understanding system.

1.2.1 Sub-lexical Linguistic Modeling

In small-vocabulary speech systems, words can be treated as basic units for speech recognition [86, 63, 80]. Each word is explicitly modeled via some clustering method. While this approach is feasible in a small vocabulary system, space and time complexities quickly become prohibitive as the vocabulary size increases. To alleviate such a problem, sub-lexical units were introduced and became popular in large vocabulary systems. Possible sub-lexical units include phones, phonemes, syllables, and others. A phone refers to the smallest perceptible discrete sound segment in a stream of speech. A phoneme refers to the smallest unit of linguistically distinctive speech sound peculiar to a given language. A syllable is a unit of pronunciation, which usually consists of a vowel, either by itself or flanked by one or more consonants (a more detailed discussion of sub-word units and their structures is given in section 2.2.3). As was mentioned in the previous section, the use of units smaller than words greatly reduces the number of acoustic models to be established, and is helpful in solving the sparse training-data problem. It also has the potential of modeling previously unseen words through the concatenation of sub-word units, satisfying the sub-word linguistic constraints. Thus, the recognizer will not be restricted to a fixed vocabulary.

The goal of sub-lexical modeling is to robustly model the construction of words using sub-lexical units. The simplest model is to establish a one-to-one mapping from a word to a sequence of sub-lexical units. For example, when phonemes are used as basic acoustic

modeling units, the phoneme sequence /iy ch/[1] is mapped to the word "each." This linear model has been used in many systems, including CMU's HEARSAY [62], IBM's TANGORA [4] and BBN's BYBLOS [17], etc. However, due to the complications of natural speech, such as the existence of heteronyms (words with the same spellings but different pronunciations and meanings), and the co-articulation effects in spontaneous speech, such a simple scheme is rarely used in modern large-vocabulary speech recognition systems. Furthermore, the necessity of handling previously unseen words demands a model other than such a simple mapping of in-vocabulary words, because canonical phoneme sequences are not known for the unseen words.

To better deal with the complex phenomena in spontaneous speech, such as different surface phonetic realizations of underlying phonemes in different phonetic contexts (unknown as phonological variations), researchers have adopted the use of pronunciation networks as sub-lexical models [24, 92]. The pronunciation network is essentially a finite state network, with each arc representing a legitimate phone. The pronunciation network models the pronunciation variations explicitly by specifying all permissible phone connections of the words in the recognition vocabulary. A path in the pronunciation network represents a legitimate pronunciation realization. The pronunciation network is usually generated from a dictionary by converting its base-form sub-lexical units to a network, then applying a series of phonological rules to the network to allow different phone realizations. Such an approach will be discussed in more detail in section 2.2.2. Some speech recognition systems, including the MIT SUMMIT [115, 36] system, use pronunciation networks for sub-lexical modeling. Notice that the pronunciation network can be weighted, by superimposing a probability model on the network. This has proved to be an effective way to model the likelihood of the alternative pronunciations supported by the rules.

One important limitation of a pronunciation network, though, is that it is a static network constructed to support a fixed vocabulary. Although the vocabulary size may be large, it is inevitable in conversational systems that the recognizer will encounter previously unseen words, i.e., words not present in its vocabulary. The ability to identify and ultimately incorporate new words on the fly is highly desirable. To relax the limitation of modeling only

[1] In this thesis, slashes are used for phonemes and square brackets are used for phones.

the in-vocabulary words and build more flexible speech recognizers, sub-lexical models can be built using solely statistical knowledge. For example, a general phone n-gram model can be used to model both in-vocabulary and out-of-vocabulary words [10]. With large amounts of training data, statistical models can capture the underlying sub-lexical phonological knowledge by learning the probabilities of different phone connections. However, since the n-gram model encodes the linguistic knowledge implicitly by probability, it is hard to analyze the linguistic features of an unknown word. Such linguistic analysis is very helpful in proposing the spelling of unknown words [19, 107], and incorporating the new words into the recognizer's vocabulary during recognition.

Studies in the seventies have revealed more detailed hierarchical structures of the sub-lexical units. In the early work of Kahn [47], the notions of describing phonological variations with larger sub-lexical units such as syllables were introduced. Later, Church [20] implemented a syllable parser using Earley's algorithm for context free grammars (CFGs). The CFG based sub-lexical modeling is a more sophisticated solution compared to the previous pronunciation network or n-gram statistical modeling approaches, since it tries to model the general hierarchical structure of words, and provides the detailed sub-lexical structural analysis for words not limited to a specific domain with fixed vocabulary. Moreover, as was mentioned above, such a detailed sub-lexical structural analysis is beneficial for proposing the spelling of unknown words and incorporating them on the fly. Probability frameworks are also introduced to augment context free grammars. For example, stochastic context free grammars (SCFGs) [14] associate each rule with a weight specifying the probability of rewriting the given left-hand-side symbol to the right-hand-side string. More details of the sub-lexical linguistic hierarchy will be discussed in section 2.3.

Note that, although context-free rules are typically used to describe the sub-lexical hierarchical structures, sub-lexical linguistic knowledge can be context-sensitive, especially at the phonetic level. For example, the word "keep" is phonemically /k iy p/, and the /k/ is aspirated (its release is followed by a burst of air). In contrast, the word "ski" is phonemically /s k iy/ but /k/ is usually not aspirated, because the /k/ is preceded by an /s/. This is a feature which native speakers in North America adopt automatically in this context. In order to model such context sensitive knowledge, researchers have proposed

probability frameworks with CFGs that account for the local context information using a data-driven approach. In particular, the ANGIE [97, 56] system developed at MIT uses *phone advancement* and *trigram bottom-up* probabilities to capture probabilistic context information through different layers of sub-lexical parse trees (to be described later in section 2.4).

Although a hierarchical sub-lexical model based on context-free grammars for structural analysis and augmented with probability models for context-sensitive modeling is powerful and flexible, it is relatively costly to implement, and may not be the optimal model under all circumstances. In conversational interfaces, sometimes very specific answers from the user are expected, for example, yes/no confirmation, transaction date, flight number, etc. In this situation, it may be more effective to use a pronunciation network based on the constrained active vocabulary, instead of using more complex probabilistic hierarchical models. Therefore, it is advantageous to have the ability to handle different sub-lexical modeling schemes under a unified framework. However, the sub-lexical model components used in most systems are specifically designed for particular types of sub-lexical models, and such components usually interact with the recognition search through customized interfaces. For example, the hierarchical sub-lexical model is often implemented as a stand-alone probabilistic parser. The parser parses the phonetic recognition hypotheses as the recognizer generates them, and sub-lexical linguistic scores for the partial theories are applied through direct interaction between the search control and the parser. Such an approach makes changing the sub-lexical modeling scheme difficult. In most cases, it is almost impossible to manipulate and create new sub-lexical models to suit the conversational context on the fly. It is our intention to formalize and apply sub-lexical linguistic knowledge within a unified framework. We will further emphasize this motivation in section 1.5.

1.2.2 Supra-lexical Linguistic Modeling

Supra-lexical linguistic modeling specifies legitimate word sequences in a particular language, with associated *a priori* probabilities. Early speech recognition systems with highly constrained tasks often use a very simple linguistic modeling approach: they typically build a word network that specifies all the possible word sequences the system needs to recognize.

While this approach is still effective in some limited-domain applications, such as a simple voice command-and-control system, it is generally not suitable for speech recognition in conversational interfaces. One obvious reason is that it is not possible to enumerate all legitimate word sequences, because the application domains are often fairly complex, and the system has to deal with spontaneous speech that usually contains a wide range of sentence patterns, including ungrammatical inputs.

For conversational interfaces, the most commonly used supra-lexical linguistic modeling approach is stochastic language modeling. The purpose of a stochastic language model is to compute the probability $P(W)$ of a sequence of words $W = w_1, w_2, \cdots, w_N$. $P(W)$ can be expressed by the chain rule as follows:

$$P(W) = \prod_{i=1}^{N} P(w_i|h_i) \tag{1.1}$$

where $h_i = w_1, w_2, \cdots, w_{i-1}$ is the word history for w_i. The probabilities $P(w_i|h_i)$ may be difficult to estimate as history grows. Therefore, h_i is often reduced to equivalence classes $\phi(h_i)$ in order to reduce the size:

$$P(w_i|h_i) \approx P(w_i|\phi(h_i)) \tag{1.2}$$

One commonly used approximation is to assume that the next word is only dependent on the $n-1$ words preceding it:

$$h_i \approx w_{i-n+1} \cdots w_{i-1} \tag{1.3}$$

The resulting model is called an n-gram language model. Most speech recognition systems use bi-gram ($n = 2$) or tri-gram ($n = 3$) models. The probabilities are trained from a large training corpus.

The advantages of using stochastic n-gram language models over simple word networks in conversational interfaces are obvious. They accept arbitrary word sequences and can provide a probability estimation for every input sentence, which is much more flexible than the highly restricted word network. Furthermore, n-gram language models are simple to use and generally yield good results when sufficient training data are available. However,

there are also limitations of this approach. For example, since the number of parameters to estimate grows exponentially with the increase in n, n-gram models can practically describe only short-distance relations between words, ignoring long-distance constraints. Moreover, an n-gram model is a flat statistical model. It may not capture the presence of important structural constituents of a sentence. It also cannot handle spontaneous speech effects such as the presence of word fragments and non-speech sounds.

In order to reduce the number of model parameters in an n-gram model, such that more robust probability estimation can be achieved with limited training data, class n-gram models are proposed, and they are widely used in speech recognition systems. In a class n-gram model, the words are mapped into word classes, and the probability of the next word is approximated by multiplying the probability of the next class given its previous $n-1$ classes and the probability of the next word given the class it belongs to. This results in a less detailed conditional context and a reduced number of model parameters. The class n-gram model can also be interpolated with the word n-gram model to achieve better adaptability to training data availability [33].

Much research has been conducted to address long-distance constraints [1, 46] and sentence structure constraints with *structured language models* [16, 76]. In these models, a formal grammar (usually a CFG) is used to describe the syntactic sentence structure, and a word is predicted not from a simple preceding word (class) sequence, but from some parse tree context. People have also been trying to combine n-gram models with formal grammars that describe the phrase structures [65, 88, 108]. Some other researchers have explored the augmentation of language modeling with both syntactic and semantic constraints [35, 110].

In general, stochastic language models are evaluated with respect to their impact on the recognition accuracy. However, they can also be assessed separately by considering how well they can predict the next word in a given text. An easy-to-compute and widely used performance measure is the *perplexity*. Perplexity is computed based on the average log-probability, LP:

$$LP = -\frac{1}{M}\log_2 P(W') \tag{1.4}$$

where W' is an M-word test corpus, and $P(W')$ is the probability of W' computed with a

trained stochastic language model. The perplexity PP is then defined as:

$$PP = 2^{LP} \tag{1.5}$$

According to basic information theory principles, perplexity is often interpreted as an average branching factor, i.e., the geometric mean of the number of words that can follow each word. A stochastic language model with lower perplexity implies that stronger constraints are imposed by the language model; thus, it may potentially be more effective in guiding the recognition search. However, one must be aware that even if perplexity is a good indicator of language model performance, it does not always correlate well with speech recognition accuracy.

1.2.3 Natural Language Understanding

The natural language component is another important constituent in a conversational interface. The system needs to understand the meaning of spoken utterances in order to give proper responses. This is accomplished by generating a formal meaning representation of the input utterance. Furthermore, the system has to deal with partial words, skipped words, non-content words in spontaneous speech, and also erroneous speech recognizer results. Typical natural language processing systems use a grammar-driven approach. A parser is used to apply syntactic constraints to the word stream obtained from the recognizer, based on a predefined grammar. After the sentence structure analysis, semantic constraints are enforced and meaning information is extracted from the resulting parse tree based on a separate set of semantic rules. In conversational systems, the sentence meaning information can be represented by a flat list of key-value pairs [38], or a hierarchical semantic frame, which provides the meaning of relevant syntactic constituents in a hierarchical structure [94, 95]. Some systems use grammar rules intermixing syntax and semantics, and semantic knowledge is encoded directly in the structural entities at low levels of the parse tree. This allows a more efficient representation, along with a straightforward method to directly obtain a meaning representation from an unannotated parse tree. More details will

be discussed in section 4.3.

The natural language component can be augmented by embedding a probability framework. For example, people have introduced stochastic grammars to augment a grammatical theory with probabilities, and assign probabilities to strings, parse trees and derivatives, as mentioned by Rens Bod [12]:

> What does a statistical extension of a linguistic theory, often referred to as a "stochastic grammar," look like? In the literature, we can observe the following recurrent theme: (1) take your favorite linguistic theory (a competence model), (2) attach application probabilities to the productive units of this theory.

There are several reasons why a probability framework is favorable. One reason is the necessity to select the preferable analysis among several hypotheses based on a large collection of training data, or to disambiguate possible confusions. Another reason is its usefulness in the so-called preference-based parsing, where it helps to prune away extremely unlikely theories, and keep the time and space complexity manageable. A third reason is that the language component augmented with probability could ideally be tightly integrated with a speech recognizer in early stages, and provide strong high-level linguistic knowledge support to enhance the speech recognizer's search.

Researchers have explored many different probability models to use with the grammar rules, such as the SCFG [14], probabilistic Tree Adjoining Grammars (TAGs) [87], probabilistic LR parsing [15, 39], and probabilistic link grammar [55]. For example, the SCFG associates a production probability with each of its context free rules. The probability of a string is defined as the sum of the probabilities of the parse trees that have the string as a yield, and the probability of a parse tree is defined as the probability of the corresponding leftmost derivation. Under the independence assumption, which assumes the current rewrite rule is independent of previous rewrite rules in the leftmost derivation, the probability of the leftmost derivation can be simplified as the production of the corresponding probabilities associated with the rules used in the derivation. The parameters for a SCFG can be estimated using the inside-outside algorithm [8].

Although SCFG provides probability support for context free grammars, it is usually not

adequate for the natural language processing needs in conversational speech interfaces. One major drawback of SCFG is its limitation of modeling linguistic phenomena in a context free formalism. However, the ability to model context-dependent linguistic information is necessary in most cases. For example, in the following SCFG model:

$$
\begin{aligned}
&\vdots \\
\text{VP} &\rightarrow \text{V NP NP} \quad (0.3) \\
\text{NP} &\rightarrow \text{pron} \quad (0.1) \\
\text{NP} &\rightarrow \text{det N} \quad (0.5) \\
&\vdots
\end{aligned}
$$

The same probability is assigned to the phrase "give me the book" and the phrase "give the book me." Researchers have proposed various probability models that can address the context sensitive information to augment the context free grammar. For example, the TINA [94] language processing component developed at MIT uses a probability model where the probability of each node in the parse tree is conditioned not only on its parent but also on its left sibling, even when the left sibling does not share the same parent. The context information as well as longer distance information are captured through probability layers in the parse tree. More detailed discussion of the probability models used with a natural language grammar will be given in section 4.4.

1.3 System Integration with Speech Recognition

Traditionally, linguistic components are used with a speech recognizer in a serially connected manner, where linguistic components simply process the results obtained from a recognizer. For example, the following "standard integration model" described by Moore [74] was used by early natural language processing systems (Boisen et al. [13], Moore [75]):

> Pick as the preferred hypothesis the string with the highest recognition score that can be completely parsed and interpreted by the natural language processor.

Such a simple integration interface is less than ideal for a number of reasons, the most significant being the following two. First, in the integration scheme described above, the natural language processor just selects a recognition hypothesis that can be completely parsed, and typically there is no probability support from the natural language component for the recognition hypotheses. As was mentioned in the previous section, a probability framework can be very helpful in assessing different recognition hypotheses, and choosing one with a preferable linguistic analysis. Second, the linguistic component cannot provide linguistic constraints in early stages of the recognition search. The natural language component merely takes in the results of a recognizer and acts as a post processor to find the sentence hypothesis that satisfies the linguistic constraints. It is usually desirable to have early knowledge integration, since the recognition search can be guided by the various knowledge sources, and partial paths not satisfying the constraints can be pruned early. For example, natural language analysis is capable of providing long distance constraints that usual n-gram language modeling is not, such as gender agreement in complex long sentences (e.g., "John is going to his house and Mary is going to hers."). Section 4.4 will address these issues in more detail.

Typical implementations of this "standard integration model" are N-best lists and word/phone graphs. In the former case, where an N-best list is used, the speech recognizer will deliver the top N full string hypotheses to the linguistic component. The linguistic component then goes through each of the hypothesized strings in order, trying to find a successful parse, and extract the meaning representation from it. In the latter case, where a word/phone graph is used as the interface of recognizer and linguistic component, the results of the speech recognizer are represented in a more compact graph format, and the linguistic component will search through the graph and find the highest scoring path with a successful parse. Both of these mechanisms are essentially loosely coupled schemes, since the recognizer takes little advantage of the constraints modeled by linguistic components. If the linguistic constraints were introduced earlier in a tightly coupled manner, the recognizer would be able to use the constraints provided by linguistic models at both the sub-lexical level and the supra-lexical level, and erroneous paths not satisfying linguistic constraints could be pruned during the recognizer's search phase. More specifically, it is advantageous

to use multiple linguistic constraints in speech recognition because, at the sub-lexical level, a proper sub-lexical model is able to handle more complex sub-lexical phenomena, such as providing detailed sub-lexical analysis, and applying sub-word structural constraints. At the supra-lexical level, on the other hand, the natural language model is able to provide longer distance constraints than a local n-gram language model, and allows more powerful mechanisms to describe natural language phenomena.

Researchers have been trying to augment the "standard integration model" and use linguistic knowledge to improve speech recognition for a long time. Zue et al. [114] used the generation capability of the natural language component to produce a word-pair language model to constrain the recognizer's search space. In the work of Ward and Issar [110], a Recursive Transition Network (RTN) was used to integrate semantic constraints into the SPHINX-II recognition system. The RTN parser is used in the A^* search portion of the decoding search, which leads to a significant reduction in understanding error rates. Murveit and Moore [78] applied Dynamic Grammar Networks to incrementally generate the grammar-state-transition table used in the standard HMM speech recognition architecture. Goddeau [39] used a probabilistic LR parser to provide both local and long-distance natural language constraints in the recognition search.

It is important to note that most efforts for tight integration of linguistic knowledge with speech recognition are specifically designed for a particular speech recognizer architecture and typically hard-wired into the recognizer. Thus, it is extremely hard to manipulate and adopt new configurations of linguistic support. Some researchers, such as Ward, Issar [110], Chung [18], and Wang et al. [108], have started to use finite state transducers (FSTs) to formalize the integration of linguistic knowledge. However, the probability models used are usually simplified and have not reached the full capacity of the originally designed linguistic systems. It is the interest of this work to propose a unified framework based on FSTs that formalizes the integration of linguistic knowledge into speech recognition and has the ability to *fully* capture the probability space defined by the original linguistic components. Some more details about FSTs will be presented in the next chapter.

In summary, we have discussed the representation of linguistic knowledge at both sub-lexical and supra-lexical levels, and the linguistic knowledge integration in speech recog-

nition. There are several key points that we would like to emphasize. First, at the sub-lexical level, most systems use context-dependent acoustic models or context-dependent rules to model phonological variations. More detailed sub-lexical linguistic knowledge such as syllable structure is typically not used. Second, at the supra-lexical level, linguistic constraints are typically provided by flat statistical language models. More detailed hierarchical supra-lexical linguistic knowledge is applied in the language understanding component after recognition is completed. Third, the interface between speech recognition and language understanding is usually a simple *N*-best list or a word-graph. It is basically a feed-forward interface, and there is little feedback from the language understanding component to speech recognition with possible hierarchical linguistic constraints. It could be advantageous to apply hierarchical sub-lexical and supra-lexical linguistic knowledge in speech recognition in a uniform way, and integrate linguistic constraints tightly in an earlier stage. The major disadvantage of tightly coupled approaches is that the search space might be considerably larger than in a loosely coupled configuration. The balance of complexity and accuracy has to be considered.

1.4 Dynamic Reliability Modeling

As was introduced in section 1.1, speech recognition can be formulated as a problem of searching for the best string of symbols, subject to the constraints imposed by the acoustic and linguistic models. In the previous sections, we have discussed the use of *linguistic* knowledge at both sub-lexical and supra-lexical levels. Another important aspect of achieving more accurate and robust speech recognition is the integration of *acoustic* knowledge, which is formulated by acoustic models. Acoustic models characterize acoustic features extracted from the input speech signal.

In typical speech recognition systems, acoustic constraints are applied uniformly across the entire utterance. This does not take into account the fact that some units along the search path may be acoustically modeled and recognized more reliably than others, due to differences in their acoustic-phonetic characteristics, the particular feature extraction and modeling approaches the recognizer chooses, the amount and quality of available training

data, and many other factors. One possible way to incorporate such reliability information is through word- and utterance-level rejection [82]. However, this approach generally provides confidence information after the recognition phase, and as such the confidence score is usually measured from a set of chosen features [48], most of which are obtained after the recognition is done. In contrast, we attempt to incorporate reliability information directly into the search phase in order to help the recognizer find the correct path. In this thesis, we also introduce our work on such dynamic phonetic reliability modeling [77], which demonstrates the possibility of integrating acoustic model reliability information tightly into speech recognition.

1.5 Thesis Motivations and Objectives

1.5.1 Motivations

The motivations of the thesis research are summarized as follows:

1. **Hierarchical linguistic constraints at both sub-lexical and supra-lexical levels are important for characterizing variability in speech in an early stage.**

   As was discussed in section 1.1 and section 1.2, at the sub-lexical level, linguistic knowledge at various sub-word levels present a comprehensive picture of the sub-lexical linguistic hierarchy. Such hierarchical linguistic knowledge is very useful in supporting generic sub-word structure constraints. At the supra-lexical level, structured language models based on a hierarchical grammar can be used to address limitations of typical n-gram language modes. Moreover, further hierarchical syntactic and semantic linguistic knowledge at the supra-lexical level can also be applied in the natural language understanding component. Therefore, hierarchical linguistic constraints are present at both sub-lexical and supra-lexical levels, and can be used in an early stage of speech recognition to help characterize variability in speech.

2. **Context-free grammars used to describe hierarchical constraints need to be augmented with context-dependent probability models.**

   In section 1.2.1, we have mentioned that hierarchical linguistic constraints are usually

described formally by context-free grammars, and augmented by probability models. The advantage of using context-free grammars is that they can model long-distance constraints and underlying structures of unit sequences. However, it is difficult to model the local context-dependency using such grammars alone. Context-dependent probability models can supplement context-free grammars by quantifying relationships applied locally to the constituents in the hierarchy among the units. For example, at the sub-lexical level, the phone realization of phonemes can be quantified by an n-gram model, which estimates the probability of the next phone based on the previous conditioning context. Such context can be represented by some hierarchical sub-word structure according to the sub-lexical context-free grammar.

3. **Speech recognition can benefit from the introduction of hierarchical linguistic knowledge.**

   There are many potential benefits of introducing *hierarchical* linguistic knowledge into speech recognition. For example, sub-word linguistic support can be provided for previously unseen words, and feedback from the natural language understanding component can be introduced to help impose supra-lexical linguistic constraints more consistently. However, we must be aware that early integration of linguistic knowledge may result in a significantly increased search space. A proper balance of accuracy and complexity is necessary.

4. **Consistent optimization procedures for integrated systems need to be developed.**

   The application of hierarchical linguistic knowledge in speech recognition involves a probabilistic parsing component to analyze the input according to the underlying grammar. This component can be implemented separately, and interacts with the recognition search using a special interface. However, such a strategy often leads to different optimization criteria and pruning procedures for the speech recognition and the hierarchical linguistic modeling components, and they have to be tuned separately. It is advantageous to have a global optimization procedure for the integrated system, which can be controlled in a uniform way.

5. **Information about acoustic model reliability can be beneficial to speech recognition.**

   As we mentioned in section 1.4, most speech recognition components apply acoustic model constraints uniformly across the entire utterance. The fact the some units are acoustically modeled more reliably than others is usually not accounted for during the dynamic search stage of recognition. It can be helpful to integrate such knowledge in conversational interfaces to improve robustness.

1.5.2 Objectives

Based on the above motivations, the main objectives of this research include the following:

1. **Develop a flexible and scalable framework integrating linguistic modeling at sub-lexical and supra-lexical levels.**

   Since both sub-lexical and supra-lexical linguistic knowledge can be used to help constrain the recognition search space, and can be formalized with hierarchical context-free rules augmented with probability models, it is appealing to establish a unified framework integrating such linguistic knowledge into speech recognition at both levels. It is also desirable to have a framework that allows linguistic models to be developed independently for different layers across the linguistic hierarchy, while providing a mechanism to integrate these models using a uniform operation. This is a parsimonious representation capable of providing the benefits of typical linguistic modeling approaches at both the sub-lexical and the natural language levels.

   As discussed in section 1.3, it might be advantageous to tightly couple linguistic components with the recognizer, such that the recognizer's search can be guided with all knowledge sources available, and linguistic components can provide feedback to the recognizer as early as possible. This can also be achieved through a unified framework that combines linguistic constraints seamlessly with acoustic constraints.

2. **Explore the feasibility of such a framework.**

   As we mentioned in section 1.3, incorporating linguistic knowledge tightly in speech

recognition may be expensive. There is a tradeoff between efficiency and accuracy. We need to explore the feasibility of constructing the proposed unified framework, and study the advantages and disadvantages of such an approach.

3. **Experiment with applications of linguistic knowledge in speech recognition.**

   We will also experiment with possible applications of integrating linguistic knowledge in speech recognition. For example, hierarchical sub-lexical linguistic knowledge can be used to model the generic structure of words, and provide a probability estimation of the likelihood that a particular phone sequence can be generated by new words that are not in the recognizer's active vocabulary. Our generalized framework also allows us to assess the contributions of different components of the hierarchy to the overall recognition task.

4. **Investigate the utility of dynamic reliability modeling.**

   Other knowledge sources can also be tightly integrated into recognition to improve its performance in a conversational interface. We will include in this thesis preliminary research on the use of dynamic reliability modeling to incorporate acoustic model reliability information.

1.6 Outline

We have discussed the general background of speech recognition in conversational interfaces, and summarized how this research is motivated and what the objectives are. Next we present an outline of this thesis. The main contents are organized into three segments. In Chapter 2 and Chapter 3 we present our research in using hierarchical linguistic knowledge in speech recognition at the sub-lexical level. Chapter 4 and Chapter 5 discuss the use of hierarchical linguistic knowledge at the supra-lexical level. In Chapter 6 we explore the use of acoustic reliability modeling. We then summarize the conclusions and propose possible future research directions in Chapter 7. The more detailed organizations within each chapter are given below:

- **Chapter 2: Sub-lexical Linguistic Modeling**

  In this chapter, we introduce the sub-lexical linguistic hierarchy, and hierarchical sub-lexical modeling with context-free grammars. Then, two types of probability augmentation to context-free grammars are discussed, context-independent rule-production probabilities and context-dependent structurally-based probabilities. Finally we introduce the use of finite-state techniques in speech recognition, and propose a framework for modeling sub-lexical linguistic knowledge using such techniques.

- **Chapter 3: Integration of Sub-lexical Models with Speech Recognition**

  This chapter concerns a study on applying hierarchical sub-lexical linguistic knowledge in speech recognition. We first experiment with hierarchical sub-lexical linguistic models with context-independent probabilities, then propose and implement an FST-based layered framework that incorporates context-dependent hierarchical sub-lexical models. Experimental results are shown for supporting previously unseen words using this framework. We also demonstrate the flexibility of this framework by examining the probability models of each sub-lexical layer, and implementing a simplified phone layer probability model.

- **Chapter 4: Supra-lexical Linguistic Modeling**

  In this chapter, we discuss the use of hierarchical linguistic modeling at the supra-lexical level. Analogous to linguistic modeling at the sub-lexical level, we first present the hierarchical approach used in natural language processing, then introduce the associated probability framework, including the stochastic context-free grammar and context-dependent probabilities. After that, we present a study for choosing effective context information using context-dependent probabilities.

- **Chapter 5: Integration of Supra-lexical models with Speech Recognition**

  We next present the idea of integrating supra-lexical linguistic knowledge into speech recognition using an FST-based layered framework similar to what is used at the sub-lexical level. After providing an overview of the FST-based framework, we present our approaches to integrating language constraints using shallow parsing with both

rule-internal and context-dependent rule-start probabilities. The FST construction for shallow parsing and rule-start probability modeling is explained in detail. Experimental results for integrating supra-lexical level linguistic knowledge using shallow parsing are also discussed.

- **Chapter 6: Dynamic Reliability Modeling in Speech Recognition**

  In this chapter, we present our preliminary work on the integration of acoustic reliability information in speech recognition. We will describe the proposed reliability models used to address the fact that, along a recognizer's search path, some acoustic units are modeled more reliably than others, due to differences in their acoustic-phonetic features, and many other factors. We will then discuss how the dynamic reliability models can be applied through dynamic reliability scoring. Experimental results and final remarks are also given.

- **Chapter 7: Summary and Future Directions**

  Finally, we outline the experiences and findings we have in this thesis, summarize the major contributions, and present possible future research directions.

Chapter 2

Sub-lexical Linguistic Modeling

2.1 Introduction

As we discussed in section 1.2, in order to increase flexibility and improve sharing of training data, acoustic models in most modern speech recognition systems are based on units smaller than words. However, high-level linguistic modeling and natural language processing in conversational interfaces typically operate at the word level. Therefore, it is necessary to bridge the gap between sub-word units and words. The goal of sub-lexical linguistic modeling is to robustly model the construction of words from sub-word units using available linguistic knowledge. For example, when acoustic models are established at the phone level, *phonological variation* (i.e., different phonetic realizations, such as flapping or glottal stop insertion, in different phonetic contexts) is an important aspect of sub-lexical linguistic modeling. Furthermore, higher level sub-lexical knowledge, such as morphology and syllabification, can help formalize and constrain generic sub-word structure, thus contributing to more effective mapping from sub-word units to words.

A fundamental issue in acoustic modeling and sub-lexical linguistic modeling is the choice of sub-word units. Generally speaking, sub-word units can be manually determined using linguistic criteria, or automatically learned from a training corpus using clustering algorithms. Examples of linguistically motivated units include syllables [27], phonemes that ignore fine phonetic details [60, 61], phones that reflect different phonetic surface realizations from phonemes [60, 58, 115], and units derived from articulatory features [29].

In contrast, the corpus-based approach does not use *a priori* linguistic knowledge, and the sub-word units are chosen by clustering acoustic data [81]. Examples of such sub-word units include *fenones* [5] obtained from direct acoustic clustering, and *multones* [3] that take into account probability distributions for different speakers and contexts. Choosing linguistically motivated units makes it easier to systematically apply sub-lexical linguistic knowledge, and is widely used in most speech recognition systems. The major drawback is that linguistically motivated units are subjective, and require linguistic expertise. Furthermore, it is difficult to manually optimize the set of linguistically motivated units for different training corpora. It is possible, however, to cluster the linguistically motivated units according to linguistic rules, training data availability, etc.

One important problem that influences the effective use of sub-lexical linguistic constraints in speech recognition is the lack of a general framework to express and apply such knowledge at different sub-lexical levels. Researchers have been trying to use separately implemented sub-lexical linguistic components, which are specifically designed for some particular recognizer architecture. They are generally difficult to re-configure or extend to suit a wider variety of application requirements. Another factor in using such sub-lexical linguistic knowledge is the integration interface with a speech recognizer. Possible interfaces include *N*-best list and phone graph hypotheses produced by the recognizer. Further linguistic constraints are applied to the *N*-best lists or phone graphs at a subsequent stage. While this approach can reduce the complexity at later stages, early integration into the recognition search phase may be desirable, because the recognition search can then be guided from available hierarchical sub-lexical linguistic constraints. Incorrect hypotheses could be pruned early in a tightly coupled system.

In this chapter, we first present some typical low-level phonological modeling approaches such as context-dependent acoustic models and pronunciation networks. Then we introduce the more sophisticated *hierarchical* modeling approach, including the ANGIE [97, 56] sub-lexical model developed at MIT, which characterizes word substructure in terms of a trainable grammar. Two possible probability augmentations of the hierarchical model are discussed, a context-independent rule production probability, and a context-dependent tree-structure based probability. After tracing the recent developments of finite-state techniques

in speech recognition, we outline the possibility of using weighted finite-state transducers to model sub-lexical linguistic knowledge, which has the potential of accommodating different types of sub-lexical models, and providing seamless integration with speech recognition.

2.2 Phonological Modeling

Phonology is a branch of linguistics that studies the sound systems of languages. It is the overall name for phonemics, phonetics and other aspects of speech sounds [112]. Phonemics is the study of phonemes, the smallest units of linguistically distinctive speech sounds peculiar to a given language. Phonetics is the study of production and perception of sounds of languages in general. Phonological variations refer to different surface phonetic realizations of underlying phonemes in different phonetic contexts. An example is given by Klatt [52] as follows:

> The phonetic transcription for the utterance: *Did you hit it to Tom?* may include the following phonological phenomena:
>
> - Palatalization of /d/ before /y/ in *did you*
> - Reduction of unstressed /u/ to schwa in *you*
> - Flapping of intervocalic /t/ in *hit it*
> - Reduction of geminate /t/ in *it to*
> - Reduction of schwa and devoicing of /u/ in *to*

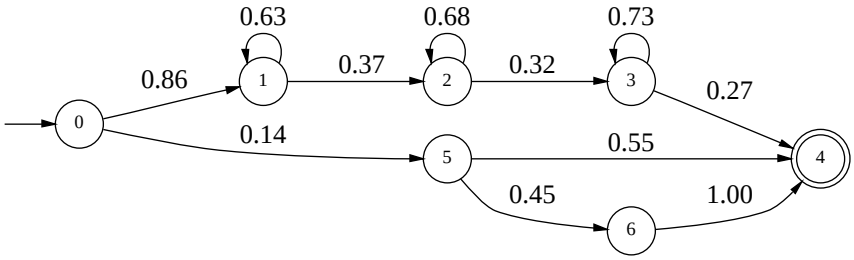
In order to achieve robust recognition, such phonological variations need to be accounted for in most systems where phones are used as basic sub-word units. This is typically handled by phonological modeling, i.e., the formalization and abstraction of phonological knowledge, and is an important aspect of sub-lexical linguistic modeling. Commonly used approaches for modeling phonological variations include implicit modeling within detailed acoustic models, and explicit modeling using pronunciation networks.

2.2.1 Implicit Phonological Modeling Using Acoustic Models

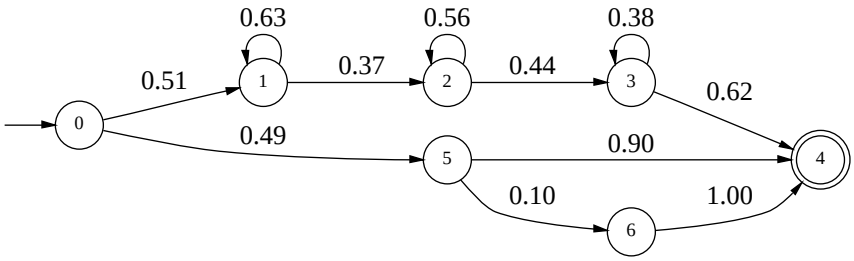
In some speech recognition systems, pronunciation variations are mainly captured by detailed acoustic models. For example, HMM based systems [59, 25] usually use some specific HMM topology to accommodate different pronunciations, then rely on large amounts of acoustic training data to learn actual phonetic realizations in different contexts. Instead of formalizing phonological variations explicitly, acoustic models are established to embed such knowledge *implicitly*. For example, in [59], Lee uses a seven-state HMM to model phones[1], and uses trained transition probabilities to reflect the different tendency of surface realizations. Figure 2-1 is given in [59] to illustrate such an approach. It is shown that the released /t/ (denoted [t]) and the unreleased /t/ (denoted [td]) are significantly different with trained HMM transition probabilities. [td] has a much higher probability (0.49) than [t] (0.14) of being just two or three frames long. Furthermore, when they are both four frames or longer, [td] has the same *closure* self-loop probability (0.63) as [t], but has much smaller *burst* self-loop probabilities associated with state 3.

One related approach of modeling phonological variations is to use context-dependent sub-word units. The idea is to design a larger set of phonetic inventory that takes into account different phonetic contexts. The sub-word units can be left-context dependent, or right-context dependent, or both. For example, a *triphone* refers to a phone with some specific left and right neighboring phones. A *diphone* is used to model the transition boundary of two successive phones. Many speech recognition systems use context-dependent sub-word units such as triphones [93], diphones [36], and context-dependent phonemes [2]. Context-dependent sub-word units can be used in conjunction with HMMs and other types of acoustic models to refine the phonological modeling process. An important factor that influences the decision of using more refined context-dependent phones or more detailed acoustic models is the availability of acoustic training data. Compared to context-independent phones, context-dependent phones usually require much more training data to obtain robust models.

[1]The basic sub-lexical units used in [59] are called phones. They are actually phoneme-like units which ignore detailed phonetic realizations, and rely on the HMMs to model phonological variations.



(A) HMM for [t]



(B) HMM for [td]

Figure 2-1: Trained HMM examples given in [59]. A seven state HMM structure is used for modeling phones, and the trained transition probabilities are shown for [t] in a syllable onset position, and [td] in a syllable terminal position. [t] tends to be realized as a closure (optional) and a released /t/, while [td] tends to be realized as a closure (optional) and an unreleased /t/.

2.2.2 Explicit Phonological Modeling Using Pronunciation Networks

A pronunciation network is a finite-state network, typically established for the active vocabulary of a recognizer [24, 115]. The finite-state network explicitly enumerates possible pronunciation variations for all words in the vocabulary. Each path (a sequence of states and transitions) in the network represents a legitimate phonetic realization of a word. While it is possible to manually specify the pronunciation alternatives in a system with limited vocabulary, it is not feasible to do so for large-vocabulary systems. Furthermore, coarticulation effects at the boundaries of two successive words make it even more difficult to construct the pronunciation network manually.

One solution of this problem is to formalize phonological variations using context-

dependent phonological rules. For example, Kaplan and Kay [49] presented a comprehensive treatment for phonological rewrite rules of the form

$$\phi \rightarrow \psi / \lambda \_\_ \rho \tag{2.1}$$

which denotes that the *target* ϕ can be rewritten by the *replacement* ψ, under the condition that the left context is λ and the right context is ρ. All the four components in the rule can be regular expressions. Such a generic form gives great flexibility in specifying contexts, targets and replacements. However, it is usually difficult to implement a system with full support of such flexibility. Many speech recognition systems use some simplified version of the generic form. For example, the pronunciation rules used in MIT's SUMMIT recognition system [42] are simplified from 2.1 by limiting the target ϕ to be a single symbol, and the left and right contexts to be a set of symbols. The following rule

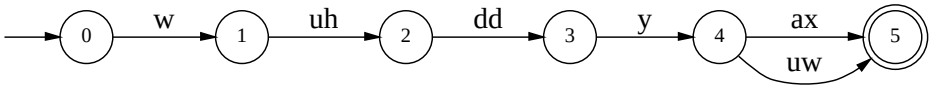
$$\{\}\ d\ \{y\} \Rightarrow dcl\ (\ d \mid jh\ ) \tag{2.2}$$

indicates that the phoneme /d/ preceding a phoneme /y/ can be realized as the non-palatalized phone sequence [dcl] [d] (closure followed by a burst), or the palatalized sequence [dcl] [jh].

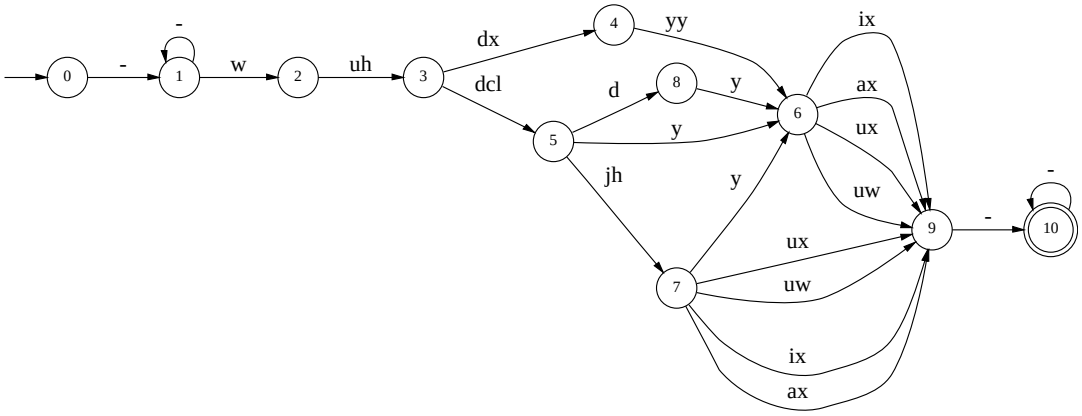
Phonological rules can be developed manually, or acquired automatically from the analysis of phonetic transcriptions of words using statistical methods. For example, the contexts used in the rules can be determined using clustering algorithms or a decision tree [6, 83].

With phonological rules, a pronunciation network can be constructed first by specifying phoneme baseform representations of each word. Then surface phonetic realizations can be generated from the phonemic baseforms by applying appropriate phonological rules in different contexts. Figure 2-2 shows a baseform phonemic network of the utterance "would you," and the expanded pronunciation network after phonological rules are applied.

Note that the pronunciation network can be weighted to further constrain phonetic surface realizations, since canonical baseforms may be realized as surface representations with different probabilities. The weights can be trained from a large collection of phonetic transcriptions, then integrated with acoustic model scores during the recognition search.



(A) Baseform phonemic network of the utterance "would you".



(B) Expanded pronunciation network after phonological rules are applied. The "-" symbol denotes silence at the beginning and end of the utterance. The phone symbols are based on ARPAbet [104], with some augmentations. For example, [yy] refers to an inter-vocalic [y].

Figure 2-2: Examples of baseform phonemic network and pronunciation network after phonological expansion.

2.2.3 Discussion

There are several advantages of explicit phonological modeling approaches such as pronunciation networks. First, since phonetic variations are explicitly enumerated, it is possible to use less detailed acoustic models. They can be robustly trained with less acoustic training data, and can be computational less expensive during recognition. Furthermore, pronunciation networks can be optimized to allow sharing of phonetic sub-word units. Thus, the recognition decoding can be quite efficient. Second, the use of linguistically meaningful phonological rules enables better understanding of phonemic and phonetic effects in speech recognition, which facilitates the study of improving recognition using available sub-lexical phonological knowledge. However, such a knowledge-based approach usually requires linguistic expertise and much human involvement, and it is costly to balance completeness and efficiency in formalization of phonological knowledge.

Although phonological knowledge is an important aspect in sub-lexical linguistic modeling, it is far from adequate. Research in linguistics has revealed detailed hierarchical structure of sub-lexical units at higher sub-lexical levels, and such knowledge also plays important roles in sub-lexical linguistic modeling. For example, morphology studies the form of words through *inflection* (i.e., creating different word forms that express grammatical information such as tense, aspect, person, number, case and agreement etc.), *derivation* (i.e., combining a morpheme affix with a word to create a new word) and *compounding* (i.e., combining two words). A word may consists of several syllables, and each syllable has a hierarchical structure built from *onset* (the initial consonant cluster), *rhyme* (including nucleus and coda) and possible *affix*. Figure 2-3 from [32] shows an example syllable template[2]. Together with phonemics and phonetics, they present a comprehensive picture of the sub-lexical linguistic hierarchy, and offer strong sub-word structural constraints that can benefit speech recognition. Furthermore, such hierarchical sub-lexical linguistic knowledge can provide rich contexts for specifying context-dependent linguistic phenomena.

Another important benefit provided by hierarchical sub-lexical knowledge is the support of words not in the recognition vocabulary. In fixed vocabulary systems, it is fairly

[2]This figure is taken from a handout of the MIT course "Automatic Speech Recognition" taught by Prof. Victor Zue and Dr. Jim Glass in 2001.

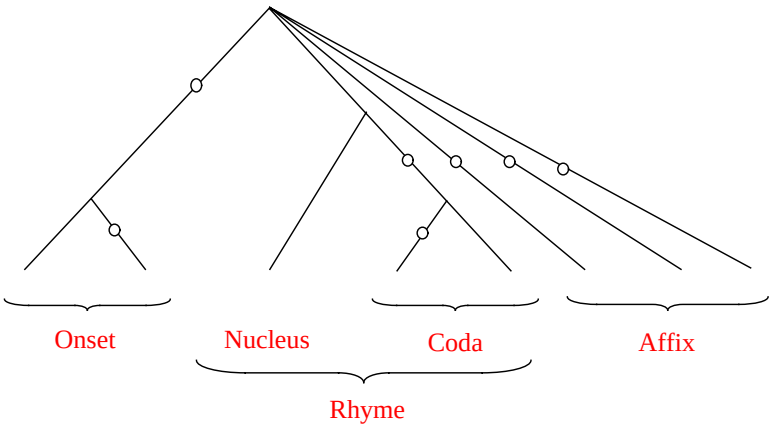


Figure 2-3: An example syllable template from [32]. Branches marked by ∘ are optional. Only the last syllable in a word can have an affix, which is highly restrictive phonemically as well.

straightforward to construct canonic phonemic baseform networks directly from a dictionary. However, conversational interfaces often encounter previously unseen words, and this is one of the major sources of mis-recognition. Proper use of the hierarchical sub-lexical linguistic knowledge at all levels can help formalize the generic sub-word structure not restricted to a specific vocabulary, thus better supporting out-of-vocabulary words. Chung [19] did an extensive study of modeling out-of-vocabulary words using hierarchical sub-lexical linguistic knowledge, and demonstrated significant improvements. In the next section, we will discuss in detail the sub-lexical linguistic hierarchy, and introduce the ANGIE [97, 56] framework which formalizes the linguistic hierarchy with context-free grammars.

2.3 Sub-lexical Linguistic Hierarchy and the ANGIE Hierarchical Grammar

As was discussed in the previous section, hierarchical sub-lexical knowledge, such as morphology, syllable structure, phonemics, and phonetics, offers powerful sub-lexical linguistic constraints. It also has the potential of providing generic structure support for previously unseen words. Researchers have been trying to formalize such sub-lexical hierarchical

knowledge through a computational model. Previous work includes a generative theory of syllables [23], phonological parsing [21, 66], morphological parsing [67], etc.

Our research on integrating sub-lexical linguistic knowledge is base on the ANGIE [97, 56] hierarchical model, which incorporates multiple linguistic phenomena at different levels of the sub-lexical hierarchy. ANGIE has been demonstrated to be successful in applying sophisticated hierarchical sub-lexical linguistic knowledge to improve robustness and flexibility of sub-lexical support for speech recognition [97, 56, 18].

In ANGIE, sub-word structure is described using a set of context-free grammar (CFG) rules. In a typical grammar, there are four abstract sub-word layers, which capture morphology, syllabification, phonemics and phonetics, respectively. The categories (non-terminals in the CFG) are grouped into separate sets for each layer, and the CFG rules are organized such that the left hand categories and right hand categories are from immediately adjacent layers. Therefore, the sub-word structure takes a highly regulated form; no layer is skipped in a successful parse. The phonetics layer includes phones that are directly associated with acoustic models. The phonemics layer includes phoneme-like units, which are used to form baseform representations of the lexicon. The syllabification layer includes syllable components, such as onset, nucleus and coda etc. The morphology layer includes morpheme-like units, such as function word, stressed and unstressed root. Such a regulated grammar is able to provide generic word structure constraints while maintaining simplicity.

Figure 2-4 shows an example sub-word parse tree of the word "introduce" using the ANGIE sub-lexical rules. Stress and onset markers (+ and !) are preserved at the phonemic level, since both of them provide additional information for modeling phonological effects at the phonetic level. Instead of using context-dependent phonological rules, phonological variations are formalized using context-free rules. One reason is that context-free rules are uniformly used through all the sub-lexical layers. Moreover, efficient parsing algorithms for CFGs are well studied. Some context-dependent information is encoded directly in the choice of the phone set, for example, the [-n] phone indicates a phone deletion immediately after an [n] phone. However, much of the context-dependent knowledge is captured by the context-dependent probability models described in Section 2.4.

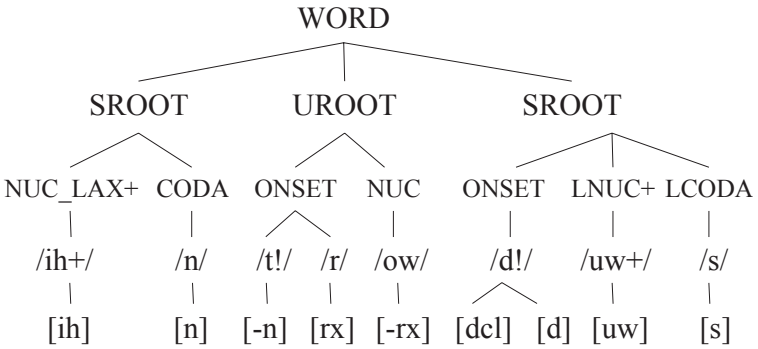


Figure 2-4: An example of ANGIE hierarchical structure for the word "introduce". Four sub-lexical layers below the WORD node are shown. The top morphology layer consists of stressed root (SROOT) and unstressed root (UROOT). The syllabification layer consists of onset (ONSET), stressed and lax nucleus (NUC_LAX+), coda (CODA), unstressed nucleus (NUC), long stressed nucleus (LNUC+), and long coda (LCODA). The phonemic layer consists of phoneme-like units with stress markers (+) and onset markers (!). The bottom phonetics layer shows the phone realizations. The deletion marker (-) encodes the phone deletion with specified left neighbor. For example, [-n] indicates that the phone is deleted following an [n] phone. Category labels are defined in the grammar, and this example is only meant to be representative of a possible labelling scheme.

2.4 Hierarchical Probability Augmentations

As was discussed above, hierarchical sub-lexical knowledge is typically formalized by CFGs. In the past, researchers have been exploring the augmentation of CFGs with probability models [14, 95]. There are several reasons to attach probabilities to a CFG in sub-lexical linguistic modeling. First, the core CFG for sub-lexical parsing is usually ambiguous. For a given input phone sequence, it is likely that there exist multiple parses consistent with the grammar. During the speech recognition search, it is necessary to evaluate the likelihood of different hypothesized phone sequences. Therefore, it is advantageous to be able to estimate from training data the parse tree probability for a particular phone sequence. Second, CFGs enforce generic linguistic structural constraints for sub-word units. Such constraints are generally tighter at higher levels of the sub-lexical hierarchy. At lower levels, a CFG by itself usually does not provide sufficient constraints because phonological variations tend to be highly context-dependent temporally. For example, stops in onset position are likely to be released. It is helpful to train phonetic realization probabilities conditioned on proper contexts, such that lower-level linguistic constraints can be modeled effectively. Third,

probability models are trained automatically, which is less costly than enumerating local context-dependency explicitly. However, typical probability models such as the n-gram models can not effectively enforce long distance sub-word structure constraints. Knowledge-based CFGs augmented with data-driven probability models may provide a good balance for modeling long distance and short distance constraints.

2.4.1 Context-independent Rule Production Probability

One straightforward method of augmenting sub-lexical CFGs with probability models is to use context-independent rule production probabilities. This is a standard approach of assigning probability estimations for different parse trees. In this case, each rule is associated with a probability of predicting its likelihood of firing given the left hand side category. It is assumed that the probability of applying a particular rule is independent of the application of any other rules. All the rule production probabilities are trained through parsing a large number of phonetic transcriptions. The probability of the whole parse tree is therefore given by the product of individual rule probabilities applied in the parse tree.

While a well-trained rule production probability model may offer a reasonable estimation of the parse tree probability, it does not take into account the context-dependent sub-lexical knowledge known to be important, especially at lower levels of the linguistic hierarchy. For example, it is hard to model the context-dependent phonological variations using context-independent rules with rule production probabilities, since the rules themselves are context-free, and furthermore, it is assumed that the rule-production probabilities are independent of each other. In Chapter 3, we will present some empirical experiments on using context-independent rule production probabilities.

Another problem of directly using a context-independent rule production probability model is that there is no sharing of similar sub-structure among rules with the same left hand category. Consequently, in order to obtain a well trained probability model, large amounts of training data must be available. One way to alleviate this problem is to approximate the rule production probability with the product of conditional probabilities of right hand categories. For example, rule-internal bi-grams can be used, which condition a right hand category on its immediate left neighboring category.

WORD								
SROOT		SROOT			UROOT			
NUC_LAX+	CODA	ONSET		NUC	ONSET		LNUC+	LCODA
/ih+/	/n/	/t!/	/r/	/ow/	/d!/		/uw+/	/s/
[ih]	[n]	[-n]	[rx]	[-rx]	[dcl]	[d]	[uw]	[s]

Table 2-1: Tabulated ANGIE parse tree for the word "introduce". The path from the top WORD node to a bottom phone node is represented by a *column*, which is used as the context condition for ANGIE's context-dependent probability models. The first column is shown in bold.

2.4.2 Context-dependent Probability

It is important to realize that at lower levels of the sub-lexical hierarchy, especially at the phonetic level, the surface realization of phones is highly context-dependent both temporally and spatially. For example, *unstressed* vowels are much more likely to be reduced to schwa, and a /k/ is more likely to be aspirated in the onset position, but tends to be unaspirated when following an /s/, as in "sky". These phonetic features are automatically inserted in such contexts by native speakers of American English. In order to model the context sensitive knowledge, researchers have proposed alternative structure-based probabilistic frameworks with CFGs, that account for context information across the boundaries of different context-independent rules.

Our research is based on the context-dependent probability models used in the ANGIE [97] framework. To illustrate the ANGIE probability model, Table 2-1 shows a tabulated form for the same parse tree as given in Figure 2-4. In this table, the path from the top WORD node to a bottom phone node is referred to as a *column* in [97]. For example, the leftmost column is (WORD, SROOT, NUC_LAX+, /ih+/, /ih/). Since ANGIE has a highly regulated layered parse tree structure, and there are a limited number of categories in each layer, it is feasible to use the entire column as the context condition.

There are two types of probabilities in ANGIE: the phone advancement probability, and the tri-gram bottom-up probability. The details are explained as follows:

- Phone advancement probability

  The phone advancement probability is the probability of the current phone (terminal) given the previous column (i.e., the rule production path from the top node to the previous phone in the parse tree). This allows effective modeling of context-dependent phonological variations at the phonetics layer, since the previous column offers rich context information across all the levels of the sub-lexical linguistic hierarchy.

- Tri-gram bottom-up probability

  The tri-gram bottom-up probability is the probability of an internal parse tree node (non-terminal) given its first child and its left sibling. This is used to construct the probability of the next column conditioned on its previous context. Although it is an approximation since we choose to use only the first child and the left sibling as conditional contexts, such a tri-gram probability allows sharing of probability of nodes with same child and sibling context; it is easier to train robustly. If the internal node is the first right-hand category of the current CFG rule, its left sibling will be a node beyond the current rule boundary. This allows modeling context-dependent information across different rules.

Given the phone advancement probability and tri-gram bottom-up probability, the probability of a parse tree is estimated by the product of the column probabilities conditioned on its left column, assuming such conditional column probabilities are independent. This is an assumption that simplifies the probability formulation. The conditional probability of a column is further defined as the product of the phone advancement probability and tri-gram bottom-up probabilities of the nodes in the right column, up to the point where it merges with the left column. For example, in Table 2-1, the conditional probability P (C_2 | C_1) of the second column C_2 given the first column C_1 is defined as:

$$\begin{aligned} P\ (\mathrm{C}_2 \mid \mathrm{C}_1) = & P\ (\text{[n]} \mid \text{[ih]},\ \text{/ih+/},\ \text{NUC\_LAX+},\ \text{SROOT},\ \text{WORD}) \\ & * P\ (\text{/n/} \mid \text{[n]},\ \text{/ih+/}) \\ & * P\ (\text{CODA} \mid \text{/n/},\ \text{NUC\_LAX+}) \end{aligned} \tag{2.3}$$

where P ([n] | [ih], /ih+/, NUC_LAX+, SROOT, WORD) is the phone advancement probability, and P (/n/ | [n], /ih+/) and P (CODA | /n/, NUC_LAX+) are the internal trigram bottom-up probabilities. The probability of the parse tree is given by:

$$P = \prod_{i=1}^{10} P\ (\mathrm{C_i} \mid \mathrm{C_{i-1}}) \tag{2.4}$$

where $P\ (\mathrm{C_i} \mid \mathrm{C_{i-1}})$ is the conditional probability of the i-th column C_i given the $(i-1)$-th column C_{i-1}. $P\ (\mathrm{C_1} \mid \mathrm{C_0})$ is the conditional probability of the first column given the *start* column marker, and $P\ (\mathrm{C_{10}} \mid \mathrm{C_9})$ is the conditional probability of the *end* column marker given the last column.

As discussed in [95], one of the central issues of designing a trainable context-dependent hierarchical probability model is to choose the context conditions. The contexts need to be specific enough to be highly constraining, while not so specific that sparse training data problems become critical. Therefore, with various applications and training data availability, it is preferable to be able to change the conditional contexts to suit different situations. For example, with limited training data, one might want to generalize the phone advancement probability context to be less specific. Instead of conditioning the probability of the next phone on the entire left column, one can use only part of the column up to a specific level. However, it is generally difficult to change the hierarchical sub-lexical linguistic probability models in speech recognition, since the implementation of probability parsing procedures is usually designed to accommodate specific probability models. Moreover, such sophisticated hierarchical probability models are often implemented in a separate sub-lexical linguistic component, and it is hard to provide linguistic constraints in early stages of recognition search. It is our intention to propose a sub-lexical linguistic modeling framework that can be seamlessly integrated in speech recognition, and allows flexible use of different hierarchical probability models. Towards explicating this goal, we will introduce the finite-state techniques in speech recognition in the next section, and discuss the possibility of sub-lexical linguistic modeling using finite-state transducers. More details of constructing and integrating context-dependent hierarchical sub-lexical models using finite-state transducers will be presented in Chapter 3.

2.5 Finite-state Techniques in Speech Recognition

A finite-state transducer (FST) is a finite-state machine that encodes a mapping between input and output symbol sequences. According to [89], an FST is defined by 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
- $E \subseteq Q \times \Sigma_1^* \times \Sigma_2^* \times Q$ is the set of edges

More basics of finite-state automata and FSTs can be found in [106, 89]. In this section, we will emphasis on the use of FSTs in speech recognition.

The transitions (edges) of a finite-state machine can be associated with weights such as probabilities, durations, penalties, etc. Since most speech recognition components and constraints can be represented by FSTs, such as the lexicon, phonological rules, n-gram language models, word/phone graphs, and N-best lists, many researchers have recently been trying to organize recognizers in the framework of FSTs. Mohri et al. [71, 72] at AT&T have been using a state-of-the-art, large-scale FST library to handle very large vocabulary speech recognition. An FST-based SUMMIT recognizer has also been developed at MIT [37].

In non-FST based recognizers, the recognition components are usually implemented separately using different data structures and control strategies, and the combination of these components is accomplished by specifically designed interfaces. In FST-based systems, however, various knowledge sources can be combined using mathematically well-defined primitive FST operations. The basic operation for combining various FSTs is called *composition* and is denoted by the composition operator $\circ$. The composition of two transducers R and S is another transducer $T = R \circ S$, which has exactly one path mapping sequence u to sequence w for each pair of paths, the first in R mapping u to some sequence v and the

second in S mapping v to w [70]. If the transducers are weighted based on probabilities, the weight of a path in T is defined by the product of the weights of the corresponding paths in R and S. For example, the recognition search space of a segment-based speech recognizer using pronunciation networks can be defined by the following FST composition:

$$S \circ A \circ C \circ P \circ L \circ G \tag{2.5}$$

where S is the acoustic segmentation, A represents acoustic models, C maps context-dependent acoustic model labels to context-independent labels, P is compiled from phonological rules, L is the lexicon, and G is the supra-lexical linguistic model, such as an n-gram language model. $P \circ L$ maps from phone sequences to word sequences, which represents the FST-based sub-lexical linguistic modeling approach using pronunciation networks. $S \circ A$ is performed dynamically when the recognizer hypothesizes segmentations of the input acoustic signal and scores acoustic models against the segments. The composition of $C \circ P \circ L \circ G$ can either be performed in advance to obtain a single FST defining the entire search space, or be performed on-the-fly, since the composition of FSTs requires only local state knowledge.

During the last decade, there has been a substantial surge in the use of finite-state methods in linguistic processing both below and above the word level. Kaplan and Kay [49] contributed to the theory of finite-state linguistic modeling. Mohri [68] [69] has recently addressed various aspects of natural language processing problems using a finite-state modeling approach. Many researchers, including Koskenniemi [53] and Karttunen et al. [50], have successfully used finite-state devices in computational morphology and other aspects of sub-lexical linguistic modeling. Context-free grammars that are prevailingly used in linguistic modeling can be represented as recursive transition networks (RTNs), which can be viewed as extensions to finite-state machines, where the set of context-free rules with the same left hand category is represented by a sub-network. CFG non-terminals are allowed as the input symbols in each sub-network. When traversing an RTN, a stack is typically used to remember the returning state before entering a non-terminal sub-network. Although RTNs can be treated as regular FSTs in implementing an FST library, not all RTNs can be represented by equivalent FSTs. The underlying language defined by RTNs can be a context-free language, which may not be represented with a finite number of states. This

problem usually does not happen at the sub-lexical level, however, since the CFGs used to describe sub-word structure tend to be highly regulated. At the supra-lexical level, certain simplifications may be necessary when constructing RTNs.

In summary, FSTs can be used to describe not only standard knowledge sources in speech recognition, such as pronunciation networks and n-gram language models etc., but also sophisticated hierarchical linguistic knowledge sources at the sub-lexical and supra-lexical levels. Therefore, it is possible to have a tightly integrated speech recognition system that seamlessly incorporates linguistic knowledge into speech recognition, and provides strong linguistic support during recognition search. Although weighted FSTs are very convenient and flexible devices, careful design considerations are needed to realize the context-dependent hierarchical probability models and other techniques used in the linguistic components.

2.6 Sub-lexical Linguistic Modeling Using Finite State Transducers

As was discussed in section 2.1, the goal of sub-lexical linguistic modeling is to robustly model the construction of words from sub-word units using available linguistic knowledge. Consequently, the fundamental objective of modeling sub-lexical linguistic knowledge with FSTs is to define a sub-word level probabilistic search space with a weighted finite-state network, which encodes such knowledge at various levels. For example, when the acoustic models are established at the phone level, this approach is to construct an FST that maps phone sequences to word sequences, with probabilistic sub-lexical linguistic constraints embedded.

The pronunciation network model can be defined by the following FST composition, which represents the sub-lexical linguistic modeling component in formula 2.5:

$$P \circ L \tag{2.6}$$

where P is the FST compiled from context-dependent phonological rules, which maps possible surface phone realizations to phonemes, and L is the phonemic baseform lexicon FST,

which maps from phonemes to words. The final composed FST can be weighted to reflect different rule application probabilities.

Similarly, the probabilistic hierarchical linguistic models discussed earlier in this chapter can be represented by the following FST composition:

$$M \circ L \tag{2.7}$$

where M encodes the sub-lexical CFG and corresponding probability models (context-dependent or context-independent), which maps from phones to phonemes, and L is the same phonemic baseform lexicon FST as in expression 2.6, which maps from phonemes to words.

To incorporate such FST-based sub-lexical linguistic models with an FST-based speech recognizer, the recognition search space is expanded by composing the sub-lexical linguistic FSTs with other recognizer FSTs that represent acoustic models and language models, etc., as shown in expression 2.5. The composition can be performed in advance, or on-the-fly during the recognition search.

2.6.1 Motivations

The motivations for using FSTs to model sub-lexical linguistic knowledge are summarized as follows:

1. **Most of the common approaches to formalizing sub-lexical linguistic knowledge can be either directly represented by FSTs, or converted to an FST under certain assumptions.**

   Some components of sub-lexical linguistic models, such as phonemic baseforms, morph lexicon, phone graphs, etc., can be directly represented by FSTs. Context-dependent phonological rules can also be effectively compiled into FSTs [42, 73] to model phonetic surface realizations. More sophisticated hierarchical sub-lexical linguistic structures are often formalized using CFGs, which can be represented by RTNs. In the case of ANGIE, for example, the grammar is highly regulated, as is the underlying language defined by the CFG. Therefore, the corresponding RTN can be compiled into an FST

with a finite number of states. Theoretically, with limited input length, RTNs can be converted into FSTs even when the underlying language is not regular. However, the resulting FST may be too large to be practical.

Furthermore, the FST can be weighted, which provides a convenient mechanism for combining knowledge-based constraints with data-driven probabilistic constraints.

2. **FST is a powerful and versatile framework to seamlessly integrate sub-lexical linguistic knowledge into speech recognition.**

   In an FST-based recognizer, the final recognition search space is defined by composing FSTs that represent various knowledge sources. During recognition, the recognizer sees a single uniform search space represented by the composed FST (static composition in advance or dynamic composition on-the-fly). This provides a flexible framework that allows independent development of different recognition components, while maintaining the tight integration in a uniform way. Moreover, the sub-lexical linguistic model itself can be further factored into several FSTs, each of which encodes knowledge at some specific level. This increases the flexibility of altering sub-lexical models at different levels to suit different application requirements. At the same time, the overall system integrity is maintained through the uniform knowledge representations.

3. **Using FSTs for sub-lexical modeling allows global optimization criteria to be applied.**

   There are mathematically well-defined optimization and standardization operations for FSTs [69], such as minimization and determinization. They can be applied to optimize the final search space uniformly. Furthermore, the pruning strategy can be easily controlled in a global scale with FSTs, since the recognition search is based on a single finite-state network. In contrast, if sub-lexical linguistic components are implemented separately and integrated with search control using customized interfaces, separate optimization and pruning procedures have to be applied, thus making it difficult to perform global optimization.

2.6.2 Challenges

There are several major challenges in using FSTs for sub-lexical linguistic modeling:

1. **Designing and constructing FSTs to represent sub-lexical linguistic knowledge.**

   In order to encode the desired sub-lexical model in an FST framework, sub-lexical modeling FSTs need to be designed and constructed. It is relatively straightforward in some cases, for example, to construct an FST to represent the phonemic baseform lexicon. However, it is less obvious in other cases, for example, to compile context-dependent phonological rules into FSTs. Much more effort is required for more sophisticated context-dependent hierarchical sub-lexical models. Although RTNs can be used to parse the phone sequences, it is difficult to incorporate *context-dependent* probabilities directly into RTNs. The main reason is that traversing an RTN requires a stack, which does not keep track of the previous sibling node context after it has been popped off the stack. It is crucial to design the proper FST architecture to fully realize the context-dependent probabilistic hierarchical constraints. In Chapter 3, we will discuss in detail our approach to constructing sub-lexical FSTs.

2. **Exploring proper application architectures using the FST framework.**

   The proposed FST framework itself is highly flexible. It is up to the user to design proper architectures suitable for different applications. For example, probabilistic hierarchical sub-lexical knowledge can be used to model both generic sub-word structure and phonological variations; thus, robust support for previously unseen words is possible. In Chapter 3, we will present example system architectures to demonstrate the applications of such an FST-based framework.

3. **Managing time and space complexity.**

   It is also a concern to manage time and space complexity. Although modern FST libraries can handle very large scale FSTs, integrating knowledge across all levels of the sub-lexical linguistic hierarchy may result in great time and space complexity. The feasibility of such an approach needs to be studied.

2.7 Summary

In this chapter, we discussed sub-lexical linguistic modeling in speech recognition. Pronunciation networks and detailed acoustic models are two typical approaches used to model low-level phonemic and phonetic linguistic phenomena. However, they are not able to support higher-level sub-lexical knowledge, such as morphology and syllabification. Further research has revealed a more comprehensive picture of sub-word structure. We introduced the sub-lexical linguistic hierarchy, and outlined the formalization of hierarchical sub-lexical knowledge with CFGs. We then discussed the motivations and procedures of augmenting CFGs with probability models, including the context-independent rule-production probabilities and structural context-dependent probabilities based on a column-column substructure.

We also introduced the finite-state techniques used in speech recognition, and presented the possibility of modeling sub-lexical linguistic knowledge using FSTs. Our goal is to construct an FST-based framework that is capable of representing context-dependent hierarchical sub-lexical knowledge, and to integrate it with speech recognition. In Chapter 3, we will present our empirical study of probabilistic hierarchical models at the sub-word level, and propose a layered approach of combining context-dependent probabilities with CFGs using finite-state transducers.

Chapter 3

Integration of Sub-lexical Models with Speech Recognition

3.1 Overview

This chapter studies the integration of sub-lexical linguistic models with speech recognition. In Chapter 2, we have pointed out that hierarchical sub-lexical linguistic knowledge can be formalized by CFGs augmented with probability models, which offer generic sub-word structure constraints as well as sub-word parse tree probability estimations. Such probabilities specify the likelihood that an input phone sequence complies with the *a priori* hierarchical sub-lexical linguistic knowledge observed from training data. We have also indicated that FST is a powerful and versatile framework, which is capable of seamlessly integrating sub-lexical linguistic knowledge into speech recognition. In this chapter, we will emphasize our efforts in modeling hierarchical sub-lexical linguistic knowledge using the FST framework as discussed in section 2.6.

Our experiments are conducted in the JUPITER weather information domain, which is described in section 3.2. We begin by exploring FST-based hierarchical models with rule production probabilities at the phoneme level, and compare it with some other approaches such as phoneme networks and phoneme n-gram models. The rule production probability model is used in this experiment because it is straightforward to integrate such probabilities into RTNs, and the RTN tools are readily available. It also allows us to examine the

effectiveness of probabilistic structure constraints above the phoneme level within the CFG formalism. However, since the low-level phonological phenomena are generally context-dependent, it is not likely that sufficient local constraints at the phone level can be offered by the rule production model.

We continue to experiment with hierarchical sub-lexical linguistic models across the full range of sub-lexical linguistic hierarchy, including morphology, syllabification, phonemics and phonetics. The limitations of rule production probabilities are shown as we extend the hierarchy to the phone level. Then we propose a layered FST modeling approach, which is capable of encoding full sub-word structure constraints and context-dependent probabilities. Details of the FST construction procedures are also presented. Next, we demonstrate the application of the layered approach in speech recognition to support previously unseen words. With this layered FST framework, it is convenient to further examine the effectiveness of the probability models for each individual sub-lexical layer, and some experimental results are given. Finally, a simplified phone layer probability model is implemented to demonstrate the flexibility of the proposed FST-based framework. We will conclude this chapter with some final remarks on modeling hierarchical sub-lexical knowledge using FSTs.

3.2 JUPITER Weather Information Domain

JUPITER [116] is a speech-based conversational interface developed at MIT that provides up-to-date weather information over the phone. JUPITER currently knows weather conditions for more than 600 cities worldwide and is designed to handle spontaneous speech from users. It is based on the GALAXY [98] client-server architecture, which serves as a platform for developing speech-based conversational interfaces. Major system components of JUPITER are implemented as individual servers under the GALAXY architecture, such as SUMMIT [115, 36] for speech recognition, TINA [94] for natural language understanding, GENESIS [9] for language generation, and back-end database servers for information access. Dialog management is also implemented as a server to control the progression of the human/computer interaction. The servers communicate with one another through a central programmable hub, which controls the information exchange using a scripting language.

This allows the hub to be conveniently specialized to a variety of different configurations.

We will mainly use the JUPITER weather information domain as the reference domain in this research, primarily because a fairly large amount of training data for JUPITER has been accumulated over the years from *real* users. The system receives an average of 100 calls per day, and the most recent corpus contains 129,485 spoken utterances. The corpus we use in our linguistic modeling experiments contains 128,461 utterances. It would be valuable to study the proper representation and utilization of linguistic knowledge in such real information access scenarios to help achieve a flexible and robust system. For example, one important challenge in JUPITER is to handle previously unseen words, such as unknown country/region/city names, unknown geography and meteorology terms, and words from out-of-domain queries. As was discussed in the previous chapter, hierarchical sub-lexical linguistic modeling has the potential to support unseen words through sub-word structure and probability constraints. The real data collected will help us identify the effectiveness and limitations of linguistic modeling approaches at different levels through the linguistic hierarchy.

3.3 Hierarchical Sub-lexical Models with Rule Production Probabilities

In this section, we experiment with probabilistic hierarchical models in the FST-based framework using context-independent rule-production probabilities. ANGIE's sub-lexical CFG is used to model the sub-word structure constraints. As discussed in section 2.5, CFGs can be directly represented by RTNs, which are treated as regular FSTs in our FST tool library. Furthermore, since the ANGIE sub-lexical grammar is highly regulated, the RTNs can also be compiled into a static FST in advance. It is relatively straightforward to use rule production probabilities with RTNs. Since rule applications are independent of one another, the production probabilities can be directly encoded by transition weights within the corresponding RTN sub-networks.

We first construct hierarchical sub-lexical models at the *phoneme* level. Sub-word structure above the phoneme level is preserved, including the morphology, syllabification, and

phonemics layers. The low-level context-dependent phonological variations are excluded from this experiment. This allows us to examine the effectiveness of sub-word structure constraints with context-independent rule production probabilities. Next, we extend this approach and construct *phone* level hierarchical models with rule production probabilities, and examine how well they can model the additional context-dependent phonological knowledge.

3.3.1 Phoneme Level Experiments

Models Investigated

We have implemented the FST-based phoneme level hierarchical model with rule production probabilities. The ANGIE sub-lexical grammar is converted to RTNs from the morphology layer down to the phonemics layer, which could provide generic sub-word structure constraints for both in-vocabulary words and previously unseen words. The RTNs are augmented with context-independent rule-production probabilities, which give probability estimations for legitimate parses of input phoneme sequences. These probabilities are trained from the JUPITER training set, and are encoded directly as the transition weights within the RTN sub-networks. Figure 3-1 illustrates the RTN topology for the phoneme level hierarchical sub-lexical model with the ANGIE grammar.

We have compared the phoneme level hierarchical model with some other commonly used modeling approaches, including a simple phoneme network model, a phoneme network with phoneme fillers, and a phoneme bi-gram model. We have also constructed a hybrid model that combines the hierarchical model with a phoneme network. All the phoneme level sub-lexical models are implemented in the FST-based framework, which can be treated uniformly when applying and evaluating them. The details of each modeling approach are given below:

1. Phoneme Network Model

This is the simplest model, and is a typical approach for most speech recognition systems with a pre-defined vocabulary. An FST is constructed to accept phoneme sequences for all known words. It provides strong constraints for acceptable phoneme sequences from words

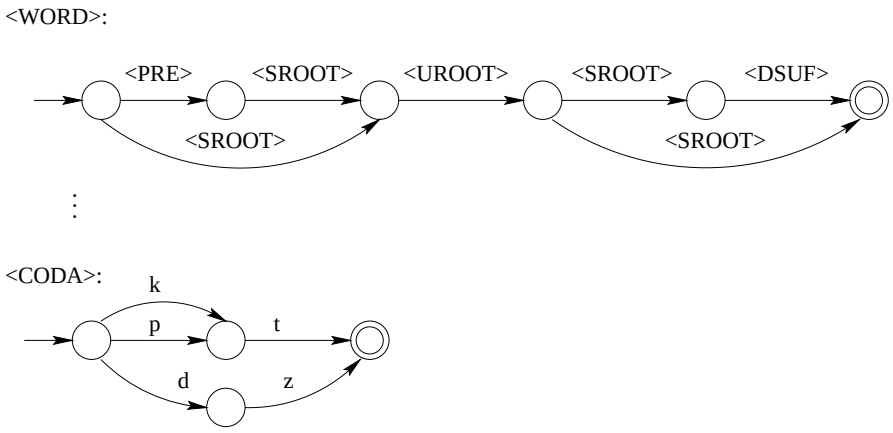


Figure 3-1: Example RTN diagram for the phoneme level ANGIE sub-lexical grammar. Each sub-network represents the CFG rules with the same labeled left hand category. For example, the top sub-network labeled <WORD> represents the CFG rules: <WORD> ⇒ [<PRE>] <SROOT> <UROOT> <SROOT> [<DSUF>], where <PRE> (prefix) and <DSUF> (derivational suffix) are optional. The bottom sub-network represent some ANGIE phonemics layer rules for <CODA>. Rule production probabilities can be encoded by transition weights within the sub-networks.

in the system's vocabulary. Therefore, the model performs well if users use in-vocabulary words only. However, since the phoneme network does not accept any novel phoneme sequences from unknown words, there may be significant performance degradation when previously unseen words are encountered. Figure 3-2 illustrates the FST topology of a phoneme network.

2. Phoneme Network Model With Phoneme Fillers

In order to allow previously unseen words in the recognizer, one approach is to use phoneme fillers to model and detect the unseen words. This idea is similar to the filler models used in a standard keyword spotting system [90, 91]. The FST has an in-vocabulary branch that defines the phoneme network for all known words in the vocabulary, and a filler branch that accepts arbitrary phoneme sequences for modeling unseen words. The same topology was proposed by Bazzi and Glass [10] and studied in detail. The two FST branches are weighted by λ and $1 - \lambda$, and the latter controls the penalty of entering the

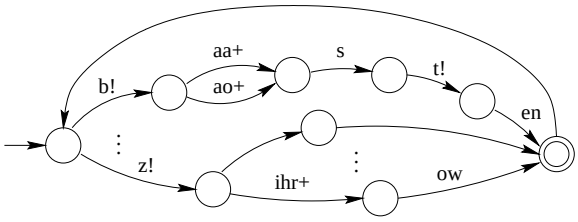


Figure 3-2: FST topology for the phoneme network model. Each path represents a legitimate phoneme sequence from words in the vocabulary.

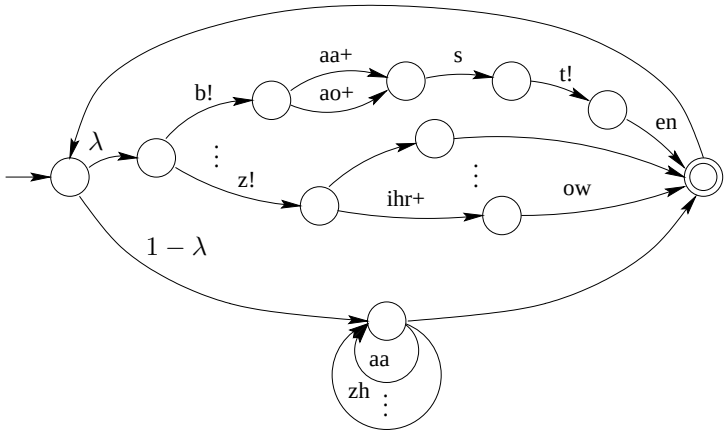


Figure 3-3: FST topology for the phoneme network model with fillers. The top branch is the same as the phoneme network model, and the bottom branch represents the phoneme fillers. λ and $1 - \lambda$ are the transition weights that control the penalty of entering each branch.

filler branch. These weights can be used to adjust the false alarm and detection rates of unseen words. In this experiment, we use a simple phoneme self-loop structure in the filler branch. Figure 3-3 illustrates this model.

Although this approach accepts unseen words and, at the same time, maintains relatively tight constraints over in-vocabulary words, the phoneme fillers do not represent any sub-lexical hierarchical linguistic knowledge. The simple phoneme self-loop structure may not be able to provide sufficient constraints for previously unseen words. The correct detection of unknown words largely relies on the accuracy and robustness of acoustic models.

3. Phoneme n-gram Model

Another possible compromise is to build sub-lexical models using solely statistical knowl-

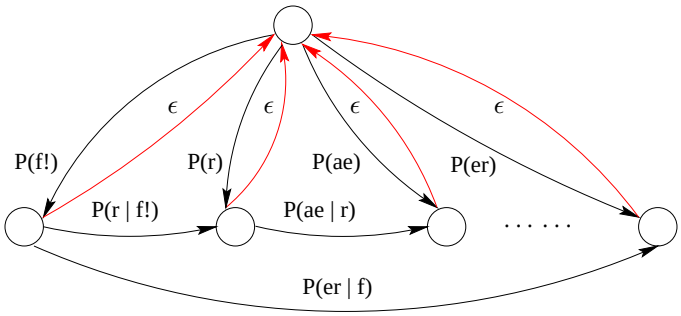


Figure 3-4: FST topology for the phoneme bi-gram model. The states at the bottom are context states, which are used to remember the current left phoneme contexts. Bi-gram probabilities are encoded in weighted transitions from current context states to next context states shown at the bottom of the figure. ϵ transitions are associated with the back-off weights, and the back-off uni-gram transitions are weighted from uni-gram probabilities.

edge. For example, we can use phoneme n-gram models to model both in-vocabulary words and unseen words. With a large amount of training data, phoneme n-gram models can learn short distance phoneme connection constraints not restricted to a particular vocabulary. Therefore, they are also able to support novel phoneme sequences from unknown words. With more detailed context information, phoneme bi-gram or tri-gram may better support unseen words than the simple phoneme self-loop structure used in the phoneme filler approach. One disadvantage of the phoneme n-gram model is that it tends to have more relaxed constraints than the phoneme network model for in-vocabulary words.

Note that n-gram models can also be converted into FSTs directly. For example, in an FST-based phoneme bi-gram model, FST states are used to remember bi-gram phoneme contexts, and the bi-gram probability is represented by the transition weight from the current context state to the next context state. To better estimate bi-gram probabilities with insufficient training data, a standard smoothing approach can be applied, which approximates the bi-gram probability using the corresponding uni-gram probability with a back-off weight associated with the current bi-gram context. This can be encoded by ϵ back-off transitions and uni-gram transitions in the bi-gram FST. Figure 3-4 illustrates the FST topology for a phoneme bi-gram model.

4. Hierarchical and Phoneme Network Hybrid Model

We also investigated the idea of combining the hierarchical sub-lexical model with the

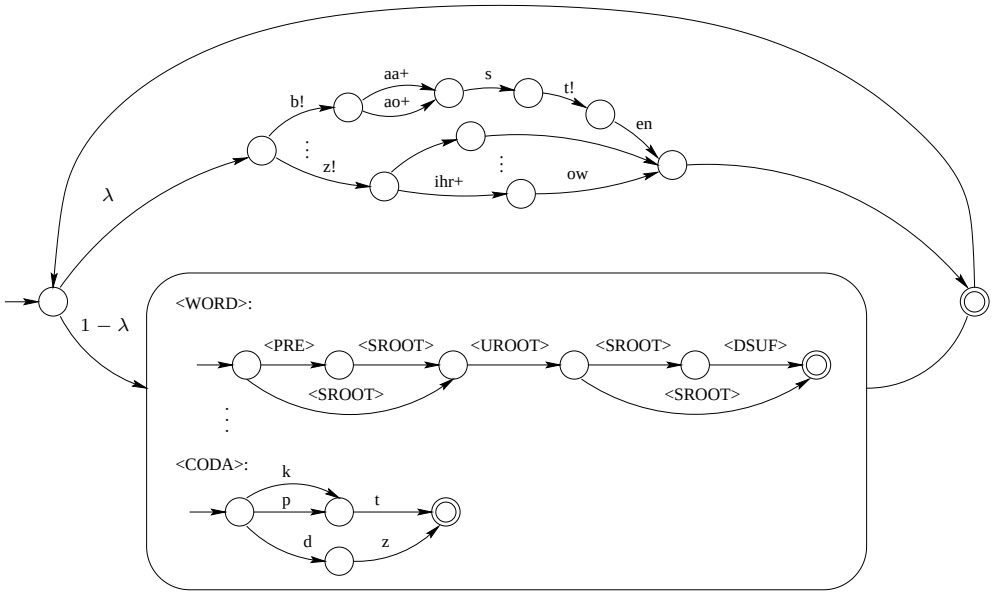


Figure 3-5: The FST topology for the hierarchical and phoneme network hybrid model. Similar to the phoneme filler model, there are two branches in the FST network. The phoneme network branch is used for modeling in-vocabulary words. The hierarchical RTN branch is used for modeling previously unseen words.

phoneme network model, which has the potential of maintaining strong in-vocabulary constraints provided by the phoneme network, as well as imposing relatively tight constraints over previously unseen words from the hierarchical sub-word structure support. Similar to the phoneme filler model, the hybrid model FST is constructed with two branches. One branch consists of a phoneme network for modeling in-vocabulary words, and the other consists of the hierarchical RTNs for modeling previously unseen words. Figure 3-5 illustrates such a hybrid approach.

Experimental Results

The phoneme level sub-lexical models described above are trained and evaluated in the JUPITER weather information domain. We have tested the sub-lexical models in terms of perplexity on a standard full test set and its subset which contains only in-vocabulary words. The utterances with artifacts, such as laugher, coughing, breathing, clipping etc. are

not included in the test sets, because linguistic knowledge generally does not apply to such artifacts, and they may be treated separately using special acoustic models [41]. The full test set consists of 1,430 utterances, and the in-vocabulary subset consists of 1,313 utterances. Out-of-vocabulary words are present in 8.2% of the utterances, and the overall out-of-vocabulary word rate is 2.4%. The perplexity results are obtained using an FST-based tool developed by us which essentially composes the test phoneme strings with different sub-lexical FSTs, searches the most likely path, and then computes the average log probability per phoneme. Table 3-1 shows the perplexity numbers for different sub-lexical models.

Sub-lexical Models	*Perplexity on Full Test Set*	*Perplexity on In-vocabulary Subset*
Phoneme Network	-	2.76
Phoneme Network with Phoneme Fillers	7.22	2.76
Phoneme Bi-gram	6.85	6.68
Phoneme Hierarchical	3.92	3.71
Hierarchical and Phoneme Network Hybrid	3.75	2.93

Table 3-1: Perplexity results of different phoneme level sub-lexical models on the full test set and its in-vocabulary subset. Both the phoneme hierarchical model and the hybrid model use context-independent rule production probabilities directly encoded in the RTNs.

From the results we see that the phoneme network model has the lowest perplexity number (2.76) on the in-vocabulary test set. However, it cannot accept any previously unseen words. Thus, it is not able to give probability estimations on the full test set for phoneme sequences containing unknown words. The phoneme network model with fillers has a high perplexity on the full test set (7.22), due to its inadequate ability to make use of sub-lexical structural information to support unseen words. The phoneme bi-gram model can model both in-vocabulary and unseen words, but it has a significantly higher perplexity on the in-vocabulary test set (6.68) than the previous two models. Compared to the phoneme bi-gram model, the hierarchical sub-lexical model with rule production probabilities reduced

the perplexity on both test sets with or without unknown words (3.92 and 3.71), due to the combination of detailed linguistic structure constraints and statistical learning from training data. For in-vocabulary words, the constraints are still not as tight as those offered by the phoneme network model, because the phoneme network is built specifically for the words in the system's vocabulary. However, it is a better balance for supporting both in-vocabulary and unseen words. Finally the proposed hierarchical and phoneme network hybrid model is able to combine the benefits and has a better overall perplexity result. Note that although perplexity is a good indicator of sub-lexical model performance, lower perplexity does not always imply better word recognition accuracy. The acoustic models we use are phone-based. If a phone-level sub-lexical model shows a perplexity improvement, it is necessary to conduct recognition experiments to verify it. In section 3.5, we will present both perplexity and recognition experiments for phone-level context-dependent hierarchical models.

In summary, we found that the hierarchical sub-lexical linguistic model with rule production probabilities is able to provide effective constraints above the phoneme level. The sub-lexical support is not restricted to a specific vocabulary because it models generic sub-word structure using linguistically motivated grammars augmented with probability estimations trained from training parses. However, when the sub-lexical hierarchy is extended to the phone level, the highly context-dependent phonological variations are not likely to be well modeled by context-independent rule production probabilities. Next, we will explore this issue by constructing hierarchical models with rule production probabilities at the *phone* level.

3.3.2 Phone Level Experiments

In order to see whether FST-based hierarchical modeling with rule production probabilities is effective at the phone level, we have constructed three sub-lexical models, including the phone bi-gram, phone tri-gram, and the phone level hierarchical model. The hierarchical model also uses the ANGIE sub-lexical grammar. In addition to the grammar rules from higher sub-lexical layers above the phoneme level, the bottom phonetics layer rules are included. Similar to the phoneme level model, all the grammar rules are compiled into RTNs, and the context-independent rule production probabilities are used. The probability

estimations are trained from a large phone transcription training set in the JUPITER domain, which consists of 83,599 utterances. A forced alignment procedure is used to obtain the reference phone transcriptions. The phone bi-gram and phone tri-gram models are trained from the same corpus.

Table 3-2 shows the perplexity results on the same two test sets as used in evaluating the phoneme level models.

Sub-lexical Models	*Perplexity on Full Test Set*	*Perplexity on In-vocabulary Subset*
Phone Bi-gram	11.19	10.91
Phone Tri-gram	5.75	5.42
Phone Hierarchical	16.71	16.54

Table 3-2: Perplexity results of different phone level sub-lexical models on the full test set and its in-vocabulary subset. The phone hierarchical model uses context-independent rule production probabilities.

From the perplexity results in Table 3-1 and Table 3-2, we see that, while the hierarchical model shows a lower perplexity than the bi-gram model at the *phoneme* level, it exhibits a higher perplexity when the hierarchy is extended to the *phone* level. For both the full test set and the in-vocabulary subset, the phone level hierarchical model is much less constraining than the phone bi-gram or phone tri-gram models. This suggests that, at the phone level, the phonological variations are highly context-dependent, and the context-free grammar with context-independent rule production probabilities is not able to model the context-dependent knowledge properly. Therefore, it becomes necessary to use *context-dependent* probabilities with the *phone* level sub-lexical hierarchical models in speech recognition.

3.3.3 Discussions

The experiments of FST-based hierarchical sub-lexical modeling with context-independent rule production probabilities demonstrated the feasibility of modeling hierarchical sub-lexical knowledge with the FST framework. Standardized FST utilities are available to manipulate and optimize the different FST-based sub-lexical models, such as composition,

best path search, determinization, minimization, etc. It is clearly beneficial to have a uniform FST framework that supports various sub-lexical models.

The hierarchical sub-lexical grammar is represented by RTNs. As we have discussed before, it is straightforward to augment RTNs with context-independent rule production probabilities. From the phoneme level experiments, we see that the hierarchical model with rule production probabilities has the advantage of being able to model generic sub-word structure, and is capable of providing tight constraints for both in-vocabulary words and previously unseen words. However, such context-independent rule-production probabilities are not able to provide sufficient support for the highly context-dependent phonological knowledge. It is necessary to explore the use of hierarchical sub-lexical models with context-dependent probabilities, such as the approach presented in section 2.4.2, in order to account for both the longer distance structure constraints and local context dependency.

3.4 FST-based Context-dependent Sub-lexical Models Using a Layered Approach

In this section, we present a novel layered approach to context-dependent hierarchical sub-lexical modeling with FSTs, and demonstrate its seamless integration with speech recognition in the FST framework. The speech recognizer we use is the FST-based SUMMIT [36] speech recognition system. With the hierarchical sub-lexical model, the recognition search space can be defined as the following cascade of FSTs, similar to the FST composition defined in 2.5:

$$S \circ A \circ C \circ M \circ L \circ G \tag{3.1}$$

where S is the acoustic segmentation, A is the acoustic model, C is the context-dependent relabelling, M encodes the hierarchical sub-lexical CFG and context-dependent probability models, L is the lexicon, and G is the language model. The entire search space can be pre-composed and optimized, or, alternatively, dynamically composed on-the-fly during the recognition search. Such a framework has the advantage of allowing tight integration of the sub-lexical models with speech recognition while preserving great flexibility for independent

development of each recognition component. Under this framework, the essential task of developing the context-dependent hierarchical sub-lexical model is to construct the FST M, such that the sub-lexical CFG and the context-dependent probability augmentations to the CFG can be embedded.

3.4.1 Related Work

Previous research towards integrating context-dependent sub-lexical linguistic knowledge into speech recognition includes the use of a separate sub-lexical parser by Lau [56], and a three-stage recognition architecture by Chung [18, 19]. In Lau's work, the ANGIE hierarchical model is implemented using a stand-alone probabilistic parser, and integrated with the recognition search control through a customized interface. This approach yields a tightly coupled system, with sub-lexical linguistic constraints applied in the early stage of recognition search. However, it is difficult to re-configure or extend the sub-lexical model to suit a wider variety of application requirements. In Chung's model, the complete recognition process is organized into three stages to accommodate multi-domain systems with flexible vocabulary. In the first stage, low-level linguistic knowledge is applied through a domain-independent phonetic recognizer. In the second stage, the phonetic network produced in the first stage is processed by parallel domain-specific recognizers. The final sentence hypothesis is selected in the third stage by a natural language understanding component. ANGIE based sub-lexical modeling is used in the first and second stages. The ANGIE sub-lexical model is represented by an FST, which is more flexible than the separate probabilistic parser approach. The FST is constructed by enumerating all parsing columns occurring in training, then connecting them using precomputed ANGIE column bi-gram probabilities, as illustrated in equation 2.3. This approach is effective and is able to capture a large portion of the sub-lexical probability space when phonetic observations are well supported by training data. It is the goal of our work to support the *full* probability space without limiting the generalization power of probabilistic sub-lexical modeling to previously unseen words, especially when sub-lexical parse trees contain columns not connected or not seen in the training data.

3.4.2 Definition of the Complete Sub-lexical Model

As was mentioned in section 2.6.2, one of the major challenges in using FSTs to model context-dependency beyond the standard CFG formalism concerns the nature of RTNs. While an input phone sequence can be parsed by RTNs according to the corresponding sub-lexical CFG, it is difficult to apply the context-dependent probabilities directly. When traversing RTNs, a stack is typically used to remember the returning state when entering a sub-network of a non-terminal. The stack cannot keep track of context information across rule boundaries. For example, the conditioning context of the ANGIE context-dependent tri-gram bottom-up probability P (CODA | /n/, NUC_LAX+) in equation 2.3 includes the left-sibling "NUC_LAX+," which has been popped off the stack and, is thus no longer available when the sub-network for the non-terminal "CODA" is entered. It is crucial to design the proper FST architecture to fully realize the context-dependent probabilistic hierarchical constraints.

In this work, we adopt a novel layered approach to capture the context-dependent probabilities. One convenient factor is that the ANGIE CFG we use produces uniform parse trees with a fixed number of layers. To construct the FST M in 3.1, we first construct RTNs from the sub-lexical CFG, which are configured to output a tagged parse string representing the parse tree for an input phone sequence. Then, a series of probabilistic FSTs are applied, each of which is designed to capture the context-dependent probabilities for a particular sub-lexical layer, and filter out the irrelevant parse tags from other layers. More specifically, M can be defined by the following equation:

$$M = S \circ R \circ L_1 \circ L_2 \circ \cdots \circ L_{N-1} \circ P \tag{3.2}$$

where N is the number of parse tree layers, S is a skip phone FST, which encodes the left context of deleted phones, R is the tagged parsing RTN, L_1 through L_{N-1} are the probabilistic FSTs for intermediate layers above the phoneme level, and P is the phone advancement probabilistic FST. The details of constructing these FSTs are given in the following sections.

M can be precomputed and optimized. It maps a phone sequence to a sub-lexical parse

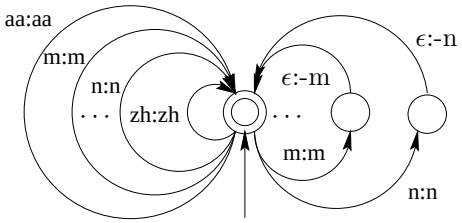


Figure 3-6: The skip phone FST diagram. The arc labeled "ϵ:-n" represents that the deleted phone comes after an [n] phone.

tree with context-dependent probabilistic support. We can choose the output of M for further processing. In our work, we output the phonemic layer categories, which can be used for further mapping to words by the lexicon FST L in FST composition 3.1.

This FST-based sub-lexical modeling design provides a view of using layered probability models to augment CFGs, and facilitates independently choosing suitable probability models for each layer. Although we will use the ANGIE tri-gram bottom-up probabilities for intermediate layers, and the ANGIE phone advancement probabilities for the phone layer, as described in section 2.4.2, this layered approach is not restricted to using such probability models as mentioned above. In section 3.7, we will demonstrate its flexibility by constructing a simplified phone layer context-dependent probability model.

3.4.3 Skip-phone and Parsing FSTs

Since the ANGIE grammar explicitly models the left context of a deleted phone in the phone set, as indicated by the phone [-n] in Figure 2-4, a skip phone FST S illustrated in Figure 3-6 is applied first to an input phone sequence. It encodes the left context of possible deleted phones. For example, the arc labeled "$\epsilon : -n$" represents that the deleted phone comes after an [n] phone. In this case, it outputs a "-n" marker for subsequent sub-lexical parsing. This constraint is encoded in the skip phone FST structure.

The skip phone FST is then composed with an RTN R, as illustrated in Figure 3-7, which is constructed from the complete *phone* level sub-lexical CFG. It is configured to output a tagged parse string representing the parse tree. For example, the first "SROOT" sub-tree in Figure 2-4 is represented by the tagged parse string "<SROOT> <NUC_LAX+> <ih+> ih </ih+> </NUC_LAX+> <CODA> <n> n </n> </CODA> </SROOT>." S

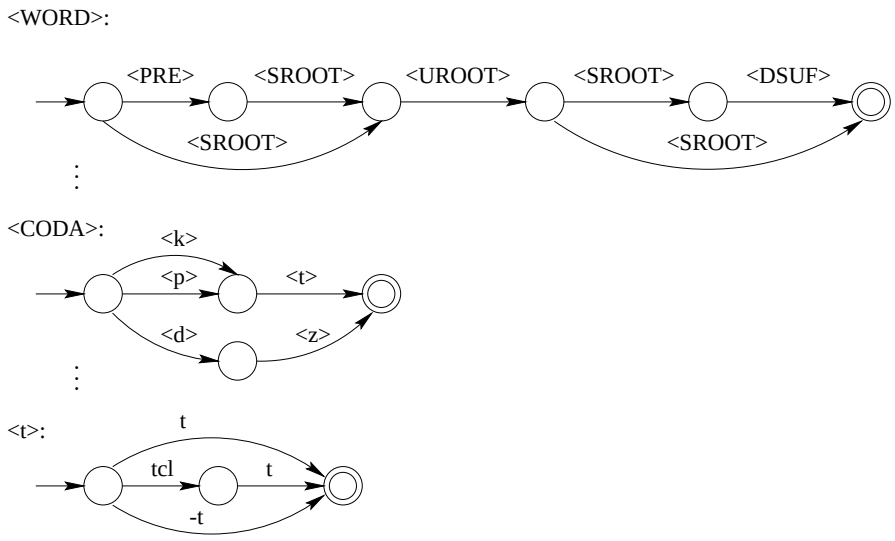


Figure 3-7: Example RTN diagram for the phone level ANGIE sub-lexical grammar. The bottom sub-network represents the ANGIE phonetics layer rules for /t/. The RTNs are configured such that the input phone sequences are mapped to tagged parse strings representing the parse trees.

and R are not weighted, and the tagged parse string is used to apply context-dependent probability models through the rest of the weighted probability FSTs for each sub-lexical parse tree layer.

3.4.4 Intermediate Tri-gram Probability Layers

In our work, the probabilities of intermediate sub-lexical layers above the phoneme level are modeled by tri-gram bottom-up probabilities as illustrated in Equation 2.3. The probability of the parent is conditioned on its left sibling and first child. The FST L_i is designed to capture this context-dependent probability for the intermediate layer i above the phoneme level. It takes in the tagged parse string, ignores irrelevant parse tags through filter transitions, and applies tri-gram probabilities of layer i. Figure 3-8 shows a diagram of the tri-gram state transitions from the current parent P and its left sibling L to its right sibling R and the parent P. The probability Prob (P | L, K) is applied during the transitions, where K is the first child of P.

Given the tri-gram state transitions defined above, the complete probabilistic FST for an intermediate layer is constructed by connecting tri-gram context states (i.e., the

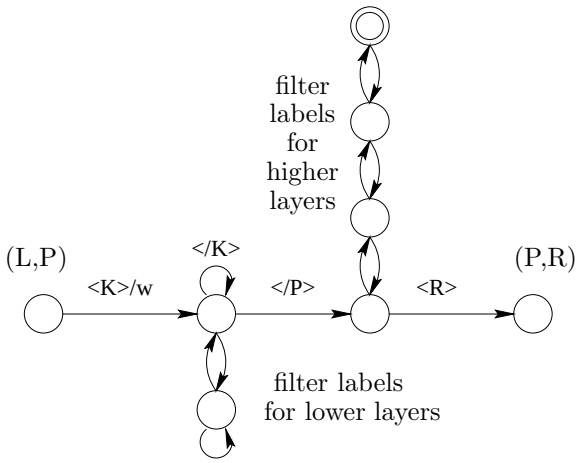


Figure 3-8: The state diagram in an intermediate layer probability FST. It shows the transitions from a state (L,P) to state (P,R), where P, L, and R are the current parent, its left sibling and its right sibling in the parse tree, respectively. "w" is the probability Prob (P | L, K), where K is the first child of P.

states remembering the transition contexts, such as the states labeled (L, P) and (P, R) in Figure 3-8) for each trained tri-gram instance. This is very similar to a standard n-gram FST (see Figure 3-4 for an illustration in the bi-gram case), except that the tri-gram FSTs used for the intermediate sub-lexical layers are constructed to follow the parse tree, and ignore the irrelevant labels not in the current layer. The filter structure could be further simplified since the categories we use are grouped into separate sets for each layer.

3.4.5 Phone Advancement Layer

The probabilities of the phone layer are defined by the phone advancement probabilities also illustrated in Equation 2.3. The probability of the next phone is conditioned on its left parse tree column. The phone layer FST P encodes such phone advancement probabilities by following the possible parse tree advancement from the previous column to the next column with a specific next phone, and applying the corresponding phone probability conditioned on the left column context. Figure 3-9 shows the possible state transitions from the left column to the right column with the same top level nodes L_0, L_1 and L_2. The phone advancement probability conditioned on the left column is encoded as a weighted arc during the transitions.

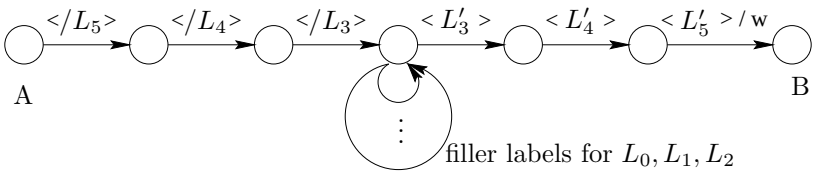


Figure 3-9: The state diagram in a phone advancement probability FST. It shows the transition from the left column A ($[L_0, L_1, L_2, L_3, L_4, L_5]$) to the right column B ($[L_0, L_1, L_2, L'_3, L'_4, L'_5]$). L_0 is the top layer category. "w" is the probability of the right phone (L'_5) given the left column.

3.5 Experimental Results

The context-dependent hierarchical sub-lexical model described above was trained and evaluated in the JUPITER weather information domain. Two probabilistic hierarchical models are constructed for evaluation. The first one uses a complete sub-lexical CFG up to the word level, and the second one uses a simplified CFG up to the syllable level, which yields a smaller sub-lexical FST M. Since the context-dependent probabilistic hierarchical sub-lexical models have the potential of providing both structure and probabilistic constraints for previously unseen words, the recognizer can be configured to output an unknown word tag for a phoneme sequence that is not in the vocabulary. We have evaluated the sub-lexical models in terms of phone perplexity on the full test set and in-vocabulary subset. The perplexity results on the training set are included as well. We have also integrated the hierarchical model with speech recognition, and evaluated the recognition performance in terms of word error rate (WER) on the full test set and the in-vocabulary subset. The baseline system uses a standard pronunciation network sub-lexical model discussed in section 2.2.2, and it does not have the ability to detect unseen words.

3.5.1 Perplexity Experiments

Table 3-3 shows the perplexity results on the training set, the full test test and the in-vocabulary subset. Compared to the test set result of the hierarchical model with rule production probabilities shown in Table 3-2, the perplexity is effectively reduced using

the hierarchical model with context-dependent probabilities, and now it becomes much lower than the phone bi-gram perplexity. This is a promising indication that the context-dependent hierarchical model can provide effective sub-lexical constraints down to the phone level. Furthermore, the perplexity gap between the training set and the full test sets is relatively smaller for the hierarchical models than for the phone bi-gram model, which suggests that the context-dependent hierarchical models might have a better generalization ability over the test data. It is also seen that the word-level hierarchical model has only slightly lower perplexity numbers than the simplified syllable-level hierarchical model. It is possible to increase the recognition search efficiency by skipping the morphology level constraints. In section 3.6, we will also see that the morphology level probability model is less effective compared to the probability models at lower levels.

We have also obtained the perplexity results for the phone tri-gram model. Although the context-dependent hierarchical models only use conditional column bi-grams, the perplexity results are close to the phone tri-gram perplexities, due to additional structure constraints and detailed conditioning contexts. Note that one factor influencing the hierarchical model perplexity results is the simplification of choosing only the best parse. The probabilities of alternative parses are ignored. One obvious advantage of the hierarchical model over the phone tri-gram model is that the hierarchical model is able to perform complete sub-lexical linguistic analysis for previously unseen words, which may be very helpful in the subsequent post-processing of the new words, such as automatically proposing the spelling, and dynamically integrating them into the active recognition vocabulary.

If a sum over all parses were computed, it would give a total probability of the phone sequence, without regard to the parse structure, as does the phone tri-gram. This would of course result in a lower perplexity than reported here. Thus, a portion of the probability space is lost due to alternate less competitive parses, and the hierarchical model perplexities given in Table 3-2 thus represent an upper bound.

3.5.2 Recognition Experiments

The context-dependent hierarchical models are integrated into speech recognition by the FST composition defined in 3.1, which supports both in-vocabulary words and previously

Sub-lexical Models	*Perplexity on Full Test Set*	*Perplexity on In-vocabulary Subset*	*Perplexity on Training Set*
Phone Bi-gram	11.19	10.91	10.18
Phone Tri-gram	5.75	5.42	5.03
Word-level Hierarchical	6.65	6.41	6.21
Syllable-level Hierarchical	6.71	6.52	6.33

Table 3-3: Perplexity results of different phone level sub-lexical models on the training set, the full test set and the in-vocabulary subset of the test set. The hierarchical models use context-dependent probabilities.

unseen words. Note that, for in-vocabulary words, the corresponding phoneme sequences are directly defined in the recognizer's vocabulary. Therefore, it is necessary to normalize the phone-level parse tree probabilities with the phoneme-level parse tree probabilities. For the unseen words, the full phone-level parse tree probabilities are used. The baseline system uses the pronunciation network model with phonological rules as defined by 2.5, which does not have the ability to detect unseen words. The recognition word error rate (WER) results are shown for the full test set and the in-vocabulary subset.

It is well known that the baseline recognizer suffers not only from the unseen words themselves, but also from the words surrounding the unseen words, because the recognizer is forced to choose a word sequence within its vocabulary that best fits the constraints. This often causes consecutive recognition errors around the unseen words. In order to better examine the effectiveness of modeling previous unseen words, two approaches of transcribing unseen words in the reference orthography are adopted while calculating the WER. In the first approach, the unseen words are left as they are. Since the recognizer with hierarchical sub-lexical models will propose a specific unknown word tag, each unknown word is counted as an error in the WER calculation even when it is correctly detected. This allows us to compare with the baseline recognizer and examine how well the added ability of modeling unseen words will benefit error recoveries surrounding an unknown word. In

the second approach, all the unseen words are mapped to the single unknown word tag in the reference orthography. Thus confusions between known words and unknown words still count as errors, but successful recognition that a word is unknown is rewarded. This approach allows us to further examine the effectiveness of unknown word detection, and is justified because knowledge of an unknown word can often be utilized in later stages of processing. For example, in a weather information dialogue system, if the natural language parser can determine a correctly identified unknown word to be a city name, the dialog manager can prompt the user to enroll the new city.

Table 3-4 shows the recognition WER results on the full test set and on the in-vocabulary subset. The two unknown-word transcribing approaches mentioned above are used to calculate the WER on the full test set for the hierarchical models.

Sub-lexical Models	*Full Test Set*		*In-vocabulary Subset*
	No UNK Mapping	*UNK Mapped*	
Pronunciation Network	15.3%	-	11.0%
Word-level Hierarchical	14.1%	12.6%	11.5%
Syllable-level Hierarchical	14.9%	13.6%	12.1%

Table 3-4: Word error rate results for different sub-lexical models on the full test set and its in-vocabulary subset. Two schemes of unknown word treatment in the reference orthography are adopted: the first one does not map unknown words to the unknown word tag, and the second one maps all unknown words to the single <unknown> word tag.

We see that, compared to the pronunciation network sub-lexical model, the word-level context-dependent probabilistic hierarchical sub-lexical model successfully reduces the error on the full test set. Without the unknown word mapping, the word-level hierarchical model is able to reduce the WER from 15.3% to 14.1% on the full test set, with a relative WER reduction of 7.8%. The reduction mainly comes from error recoveries around the unknown words. If the unknown words are mapped to the specific <unknown> word tag, the WER further drops to 12.6% with a relative WER reduction of 17.6%, which indicates successful

unknown word detection. The word-level model has a relatively small degradation in word error rate on utterances that contain only in-vocabulary words, because the pronunciation network is constructed specifically for the system's vocabulary, without unknown word support. We can also see that the word-level model yields a better performance than the syllable-level model, since it has stronger structure constraints at the high-level morphology layer. However, the word-level model FST has 60.4K states and 599K arcs, while the syllable-level model FST has only 42.1K states and 462K arcs. Thus the syllable-level model results in a more efficient decoding process.

3.6 Layer-specific Probability Models

The proposed solution for context-dependent sub-lexical modeling using the layered approach decomposes the sub-lexical FST into parsing RTNs and a series of layer-specific probabilistic FSTs, as defined in equation 3.2. This provides a layered view of constructing the hierarchical probability models. One immediate benefit of using this layered framework is that it is very convenient to examine the context-dependent probability models for each sub-lexical layer, which is difficult to do with usual stand-alone probabilistic parsers. This would facilitate choosing a suitable probability model for each sub-lexical layer.

We have conducted preliminary research on layer-specific probability models. In order to study the effectiveness of the probability models for each layer, we have substituted each layer-specific context-dependent probability model with a normalized flat probability model, which assigns an equal probability for every node with some specific context. Such a flat model essentially eliminates the context-dependent probabilistic constraints for that specific layer. The context-dependent models for the rest of the layers are left intact. By comparing the resulting models with the original full model, we are able to tell how effective the probability model for that specific layer is. Figure 3-10 shows perplexity results on the full set and the in-vocabulary subset for the original full model, and the models with a flat probabilistic model for a specific layer, including the morphology, syllabification, phonemics and phonetics layers.

The results clearly indicate that the context-dependent models are more effective at

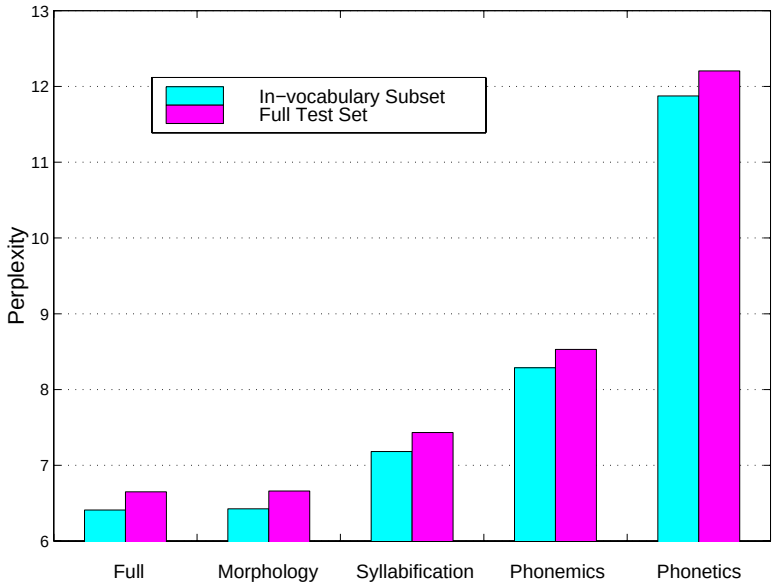


Figure 3-10: The perplexity results in the *absence* of context-dependent probability models for specific layers. The perplexity numbers are shown for the full test set and the in-vocabulary subset. The leftmost results are for the full context-dependent hierarchical model baseline.

lower levels of the sub-lexical linguistic hierarchy. For example, without the phonetics layer context-dependent probability models, the perplexity on the full test set jumped to 12.2, compared to the original model perplexity of 6.65. Similar results are shown for the in-vocabulary subset. This is consistent with the fact that the phonological variations are highly context-dependent, and the context-dependent probability models beyond the standard CFG formalism are essential at the phonetics layer. Since the context-dependent probability models are not as effective at higher levels, especially at the morphology layer, it may be possible to simplify the higher level probability models with rule-based context-independent models, which can be directly encoded within the RTN sub-networks.

3.7 Simplification of Phone Layer Probability Model

Another benefit of using the layered FST framework for hierarchical sub-lexical modeling is the flexibility of tailoring probability models for each sub-lexical layer, while preserving the

overall integrity of the sub-lexical model. There are several advantages of using layer-specific probability models. First, as shown in section 3.6, the effectiveness of the probability models varies at different levels. Therefore, it would be beneficial to be able to treat them separately. Second, with different applications and different amounts of training data available, one may want to adjust the probability models for each layer to suit the particular situations. The layered framework is able to deliver such flexibility with little cost.

To demonstrate the flexibility of the proposed layered FST framework, we have constructed a hierarchical sub-lexical model with simplified phone-level probabilities. Instead of using the entire left column as the conditioning context, we choose to only use the left sibling phone and the left parent to represent the left context. The original phone advancement probability FST is replaced by another FST to encode such new conditional probabilities. Since the simplified phone advancement probabilities are like the intermediate layer tri-gram probabilities, similar FST construction procedures can be used. Table 3-5 shows the size comparison of the original and simplified phone level FSTs after applying standard optimization. As we can see, the simplified phone advancement probability FST has a much smaller number of states and arcs.

Sub-lexical Models	*Number of States*	*Number of Arcs*
Original Phone Advancement Probability Model	54.8K	235K
Simplified Phone Advancement Probability Model	24.6K	143K

Table 3-5: Phone layer FST size comparison for hierarchical sub-lexical models with the original and simplified phone advancement probabilities.

The disadvantage of consolidating the column contexts is that it relaxes the phone level probabilistic constraints, if sufficient training data are available. However, if the lack of training data is of a major concern, this approach would allow better sharing of phone probabilities with similar left contexts, thus alleviating the sparse data problem. We have evaluated the simplified phone layer model in terms of perplexity and recognition WER on

the full test set and in-vocabulary subset. Table 3-6 shows the perplexity results. We can see that, with the reduced FST size, the simplified model has suffered some degradation in perplexity. It indicates that, in this case, the training data are sufficient to train relatively robust phone-level probability models with full left column context.

Sub-lexical Models	*Perplexity on Full Test Set*	*Perplexity on In-vocabulary Subset*
Original Phone Advancement Probability Model	6.65	6.41
Simplified Phone Advancement Probability Model	8.02	7.73

Table 3-6: Perplexity results of hierarchical sub-lexical models with original and simplified phone advancement probabilities on the full test set and its in-vocabulary subset. The hierarchical models use context-dependent probabilities.

Table 3-7 gives the recognition results. Compared to the original phone advancement probability model, the simplified model performs slightly worse on both the full test set and the in-vocabulary subset, which is consistent with the perplexity results above. This experiment shows the feasibility of applying different context-dependent probability models at the phone level. It also demonstrates the flexibility of the layered FST framework to accommodate different probability models for each sub-lexical layer. Although we have mainly used the ANGIE context-dependent probability models in this research, it is possible to apply other probability models using the same layered FST approach. It is a generic framework not restricted to any particular probability models, which is one of the major advantages over some other probabilistic parsing approaches that are designed to use pre-defined probability models.

3.8 Discussion

In this chapter, we have first demonstrated the effectiveness of generic structural constraints provided by hierarchical sub-lexical modeling above the phoneme level. Linguistically motivated structural constraints are not restricted to a particular vocabulary; therefore, they

Sub-lexical Models	*Full Test Set*		*In-vocabulary Subset*
	No UNK Mapping	*UNK Mapped*	
Original Phone Advancement Probability Model	14.1%	12.6%	11.5%
Simplified Phone Advancement Probability Model	14.5%	12.9%	11.9%

Table 3-7: Word error rate results of hierarchical sub-lexical models with original and simplified phone advancement probabilities on the full test set and its in-vocabulary subset. The same unknown word treatment schemes are used.

can be used to support both in-vocabulary words and previously unseen words. Compared to some other models with the ability to support unseen words, such as the phoneme network with filler models and the phoneme bi-gram model, the hierarchical model is able to provide relatively tight constraints for phoneme sequences from unknown words. We have also shown the necessity of using context-dependent probability models to augment the CFG, especially at low levels of the sub-lexical linguistic hierarchy. With the ANGIE context-dependent probabilities, the phone level hierarchical model reduced perplexity substantially compared to the context-independent rule production probabilities.

There are many ways of attaching context-dependent probabilities to CFGs. The central issue is to design a probability model with effective conditioning contexts. With different applications and training data availability, it is advantageous to have a flexible framework that can accommodate different probability models. It is also desirable to have a tight integration interface with speech recognition, which can help impose sub-lexical linguistic constraints early in the recognition search. With these goals, an FST-based framework with a layered approach is proposed to construct context-dependent hierarchical sub-lexical models, which can be seamlessly integrated with speech recognition in a unified FST architecture. With the ANGIE probability model, the layered framework is able to catch the full ANGIE sub-lexical probability space with little sacrificing of its generalization capability. Moreover, the probabilistic models for each layer are not restricted to the ANGIE tri-gram and phone advancement probabilities, and it is also convenient to conduct research on layer-specific models using this framework.

Our recognition experiment shows successful support of previously unseen words using context-dependent hierarchical sub-lexical models, and demonstrates the feasibility of implementing such models within the proposed layered framework. One advantage of modeling unseen words with hierarchical linguistic models is that the complete sub-lexical linguistic analysis is available for subsequent post-processing. For example, such linguistic information can be used to propose the spelling of unknown words, and dynamically incorporate them into the recognition vocabulary. A more detailed study in this direction is given by Chung in [19].

Note that the hierarchical sub-lexical modeling approach with the CFG formalism and probabilistic augmentations is also commonly used in the natural language understanding component of speech understanding systems. Therefore, we are interested in incorporating a similar layered FST-based framework at higher supra-lexical linguistic levels as well. It would be beneficial to derive consistent supra-lexical constraints from the natural language component and incorporate them with speech recognition. One difficulty, however, is that the grammars used at language levels are usually much more complicated compared to the regulated sub-lexical grammars. It may be necessary to simplify the supra-lexical structures and probability models in order to construct manageable FSTs. In the next two chapters, we will discuss hierarchical linguistic modeling approaches above the word level, and explore a unified framework with layered FSTs for both sub-lexical and supra-lexical linguistic modeling.

Chapter 4

Supra-lexical Linguistic Modeling

4.1 Introduction

Supra-lexical linguistic modeling in speech recognition refers to formalizing and applying linguistic knowledge above the word level. In addition to acoustic knowledge and sub-lexical linguistic knowledge, supra-lexical linguistic knowledge is another major knowledge source constraining the recognition search space. As we know, given a particular language, there are certain linguistic rules governing the sentence structure and meaningful word connections in the language. The fundamental task of supra-lexical linguistic modeling is to formalize such *a priori* knowledge of the language, and use it to guide the recognition search. Typical supra-lexical linguistic modeling approaches used in speech recognition include the basic n-gram language model and some of its variants, such as the class n-gram model. Researchers have also explored other types of models to address the limitations of the n-gram model. For example, structured models have been proposed, which impose sentence structure constraints using a formal grammar.

Another component in a speech understanding system that makes heavy use of supra-lexical linguistic knowledge is the natural language understanding component. The objective of natural language understanding here is to generate formal meaning representations from sentence hypotheses given by speech recognizers, as illustrated in Figure 1-2. One approach is to parse the input sentence based on hierarchical natural language grammars, then perform semantic analysis based on the resulting parse tree. There are several important

challenges for natural language understanding in a speech understanding system. First, it has to deal with erroneous speech recognition hypotheses. Furthermore, even if the speech recognition results were perfect, there could be a lot of ungrammatical phenomena in spontaneous speech that violate the predefined grammar. These factors have to be considered while applying linguistic knowledge in language processing.

In this chapter, we discuss the use of supra-lexical linguistic knowledge in both speech recognition and language understanding components in a speech understanding system. It is important to point out that, although supra-lexical linguistic knowledge is used in both speech recognition and natural language understanding, the constraints applied are not necessarily consistent, because they are usually modeled separately using different approaches in the two components. We will address the problem of deriving consistent constraints in Chapter 5. In the following sections, we first discuss supra-lexical linguistic modeling approaches for speech recognition. After introducing some basic unstructured supra-lexical linguistic models, we present the structured models with the use of formal grammars. Then, hierarchical natural language understanding approaches for spoken language systems are discussed, including the MIT TINA [94] natural language system, which is used in our research of supra-lexical linguistic modeling. The probability models used with natural language systems are also examined, with the emphasis of context-dependent hierarchical probability models. Next, we elaborate on our research on choosing effective conditional context for the context-dependent hierarchical probability models. The experimental results and analysis are also included. Finally, we summarize the important issues we have discussed for supra-lexical linguistic modeling.

4.2 Supra-lexical Linguistic Modeling in Speech Recognition

In highly constrained speech recognition tasks, it is possible to directly specify all possible word sequences that need to be recognized. These sentences are usually grammatically sound, and users are required to follow them exactly. Even in such a constrained situation, however, it is often arduous to enumerate all the sentences explicitly. Therefore, a commonly used approach is to build a word graph that accepts the sentences of interest. Certain

common parts of the sentences can be shared in the graph, resulting in a more compact and efficient representation. Since only the word sequences specified by the word graph will be accepted, such an approach eliminates a large portion of the recognition search space with hard decisions. Figure 4-1 shows an example word graph that accepts flight numbers.

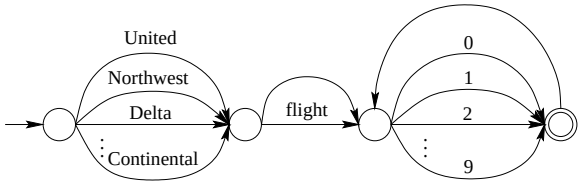


Figure 4-1: Simple word graph accepting a flight number with arbitrary digit length.

While the requirement of complying with the designated word graph may be effective in constrained tasks, they are clearly not flexible enough to handle spontaneous speech in more complex domains. For speech recognition in conversational interfaces, the common approach for supra-lexical linguistic modeling is to apply statistical n-gram language models and some of the n-gram model variants, such as the class n-gram models. These models learn local word connection constraints through a large training corpus, and associate each word sequence with a probability estimation indicating the likelihood that it would occur in the particular language. Unlike the simple word graph approach, there are no hard decisions made to rule out any specific word sequence. Instead, low probabilities are assigned to unlikely word sequences. Therefore, it is much more flexible than the simple word graph. However, the benefit of being more general often comes at the price of being less constraining.

The basic n-gram models are unstructured, and the n-gram probabilities are trained to capture only local context-dependencies. Ideally, one can capture progressively longer distance constraints by increasing n. With an increase in n, however, it becomes much more difficult to train a robust model with limited training data. Therefore, n-gram models are generally not able to model long-distance relationship between words. Furthermore, the presence of important structural constituents of sentences are not accounted for by the flat n-gram models. Many researchers have been exploring structured statistical language

models to address these problems. The idea is to impose some structure constraints over the word sequences, usually formalized by linguistically motivated formal grammars. The major practical problem of using such grammars is that they may only provide a partial coverage of natural languages. Therefore, it would be useful to combine n-gram models with grammars such that a better balance of coverage and precision can be achieved.

In this section, we will discuss in more detail the commonly used n-gram models, including some of the variants, and the structured language models.

4.2.1 n-gram Model and Its Variants

1. Basic n-gram Model

The basic n-gram model is widely used in speech recognition systems, mainly because it has flexible coverage, and can offer reasonable constraints in many circumstances. As described in section 1.2.2, it provides a probability estimation of a word sequence $W = w_1, w_2, \cdots, w_N$, which is approximated by the product of the conditional probabilities of each word conditioned on the previous $n-1$ words. For example, the bi-gram model ($n = 2$) is defined as follows:

$$P(W) = \prod_{i=1}^{N+1} P(w_i|w_{i-1}) \tag{4.1}$$

where w_0 is the start of sentence marker, and w_{N+1} is the end of sentence marker. $P(w_i|w_{i-1})$ is the probability of the next word given its previous word, which is trained by simply counting the word-pair occurrences in a large training corpus.

One of the major concerns of the basic n-gram model is the sparse data problem. With an increase in n, the number of model parameters increases exponentially, and many possible n-gram combinations have never occurred in the training corpus. One way to deal with this problem is to introduce smoothing techniques. Such techniques try to estimate the high order n-gram probability with lower order probabilities, while maintaining the integrity of the probability space. For example, if the word pair $(w_1 w_2)$ has never appeared in the training corpus, the back-off bi-gram [51] estimates the probability of $P(w_2|w_1)$ as follows:

$$P(w_2|w_1) = q(w_1)P(w_2) \tag{4.2}$$

where $P(w_2)$ is the uni-gram probability of w_2, and $q(w_1)$ is a normalization factor chosen so that $\sum_{w_2} P(w_2|w_1) = 1$.

2. Class n-gram Model

Another method of dealing with the sparse data problem is to reduce the number of model parameters. The typical approach is to use class n-gram models, which apply less detailed conditional contexts compared to the word n-gram models. For example, the class bi-gram model maps words $W = w_1, w_2, \cdots, w_N$ into equivalence classes $c_1, c_2, \cdots, c_n$, and the probability of the word sequence W is approximated by:

$$P(W) = \prod_{i=1}^{N+1} P(c_i|c_{i-1})P(w_i|c_i) \tag{4.3}$$

The words can be manually clustered into meaningful classes. For example, all the city names of a particular state may be put in one class. They can also be clustered automatically with some clustering criteria. For example, words with similar statistical behavior can be grouped together. Furthermore, the number of classes can be adjusted to suit the training data availability.

It is easier to obtain more robust probability estimation for class n-gram models with less training data. At the same time, the advantage of flexible coverage from n-gram models is preserved. Therefore, class n-gram models have been widely used in speech recognition systems.

3. Predictive Clustering

As illustrated in equation 4.3, the regular class n-gram models approximate the probability of a word w_i given its history h_i by the following factorization:

$$P(w_i|h_i) \approx P(c_i|c_{i-1}, c_{i-2}, \cdots)P(w_i|c_i) \tag{4.4}$$

where the first term $P(c_i|c_{i-1}, c_{i-2}, \cdots)$ represents the probability of the next class c_i given

its previous class contexts, and the second term $P(w_i|c_i)$ is the probability of the next word w_i given the class c_i. It assumes the probability of w_i within its class is independent of previous class contexts. Instead of using such an approximation, predictive clustering [40] takes into account the class history when estimating the probability of the next word, as defined by equation 4.5. Although the models become larger, it has been shown that predictive clustering can produce good results when combined with pruning [40].

$$P(w_i|h_i) = P(c_i|c_{i-1}, c_{i-2}, \cdots)P(w_i|c_i, c_{i-1}, \cdots) \tag{4.5}$$

Note that the classes can be represented by a trivial CFG. With regular class n-gram models, the probabilities of the words are defined within the class rules. In the predictive clustering case, the probabilities of the words are defined across the class rule boundaries. This shares some similarity with the sub-lexical context-dependent models discussed in section 2.4.2. We will also see its similarity with context-dependent hierarchical probability models for natural language understanding, which will be discussed in section 4.4.3.

4.2.2 Structured Language Models

The basic n-gram models mainly capture local word connection constraints. In order to better model long-distance constraints and structural constituents of a sentence, *structured language models* [16, 76] are introduced. In these models, a formal grammar (usually a CFG) is used to describe the syntactic sentence structure, and the next word is predicted not from a simple preceding word (class) sequence, but from some parse tree context. For example, the probability of a word sequence $W = w_1, w_2, \cdots, w_N$ and a complete parse tree T can be calculated as follows [76]:

$$P(W,T) = \prod_{i=1}^{n} P(w_i|t_{i-1})P(t_i|w_i, t_{i-1}) \tag{4.6}$$

where t_i is the i-th incremental partial parse tree context covering the words w_1 through w_i. Similar to the basic word n-gram model, with limited training data, it becomes necessary to classify the conditional parse tree contexts to alleviate the sparse data problem. For example, in [16], a partial parse tree is represented by its *exposed head*, which is the root word

node with its part of speech (POS) tag. The partial parse trees can also be clustered using a hierarchical clustering method [76]. Other representations of the parse tree contexts are also possible. For example, the *column* context used in the ANGIE context-dependent hierarchical sub-lexical model discussed in section 2.4.2 can be viewed as another form of parse tree context abstraction. In section 4.4, we will see some other probability frameworks with formal grammars that are used for natural language understanding, such as the stochastic CFG, and context-dependent hierarchical models.

Note that the syntactical grammars applied in the structured language model can only cover the word sequences that comply with the predefined grammar. On the one hand, the use of grammars can provide tight structural constraints for grammatical sentences. On the other hand, however, such an approach can be problematic in conversational interfaces, because many speech inputs from users contain ungrammatical utterances. Partial phrases, skipped words, disfluencies and other spontaneous speech phenomena can result in failure of a complete parse according to the grammar. One way to address such a problem is to combine structured models with n-gram models, so that sufficient generality (i.e., coverage) and tight constraints (i.e., precision) can be better balanced. Furthermore, it is possible to capture both long-distance constraints and local context-dependency using the combined approach.

One example of the combined model is the phrase class n-gram [65]. In this model, a CFG is used to describe phrase structure. The sentence is reduced to a sequence of phrase tags and words that are not covered by the phrase grammar. The reduced sentence is modeled by a regular n-gram model, and the probability of the original word sequence is defined as the product of parsing and n-gram probabilities. This is analogous to the class n-gram, with the class mapping replaced by phrase level hierarchical structures. Compared to the structured language model that requires a complete parse, such a combined model can capture the phrase fragments through a phrase grammar, and keep the top layer coverage flexibility by using the n-gram model. Figure 4-2 illustrates such phrase-level reduction.

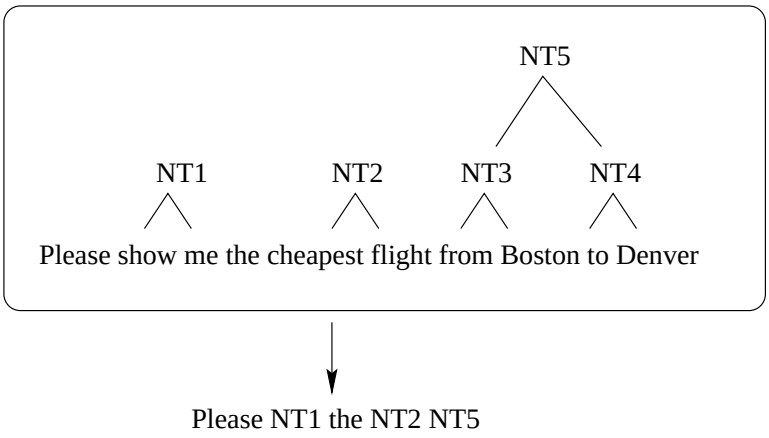


Figure 4-2: The phrase reduction from the original sentence to the reduced sentence. NT1 through NT5 are nonterminals in the phrase grammar.

4.3 Hierarchical Natural Language Understanding

4.3.1 Overview

Within the framework of conversational interfaces, the natural language understanding component performs syntactic and semantic linguistic analysis for sentence hypotheses proposed by a speech recognizer. More specifically, it parses the input recognition hypotheses, selects the best candidate, and generates a formal meaning representation that can be further interpreted for appropriate action.

Generating a formal meaning representation requires the analysis of hierarchical meaning-carrying structure in a sentence. Unstructured statistical language models, such as the n-gram model, are generally not appropriate for natural language understanding, because they do not capture meaningful structures through a structured analysis. Such analysis is usually achieved by parsing the input sentence according to a hierarchical natural language grammar, which is often written in the form of CFGs. Unlike the CFGs used in structured language models for speech recognition, which typically focus on syntactic structures, the natural language grammars used in speech understanding usually consider the semantic aspects of the language as well.

There are several factors that need to be considered when designing the natural language

component for a speech understanding system. First, it is desirable to realize high coverage of well-formed sentences, such that complete linguistic analysis can be conducted. Second, ill-formed sentences need to be rejected, which can help the selection of the best recognition candidates. Third, in the presence of recognition errors and ungrammatical speech inputs, it is important to perform partial analysis, and recover as much information as possible. These factors are inter-dependent, and improvement in one direction often results in degradation in another direction. Therefore, it is necessary to be able to balance them properly.

In this section, we first discuss the natural language grammar used in natural language understanding systems, then outline the TINA natural language system, which is used as a basic framework for our supra-lexical linguistic modeling studies. Several important features of TINA are also included, such as robust parsing, feature unification and the *trace* mechanism.

4.3.2 Natural Language Grammar

There are many grammar formalisms used in classic natural language processing. For example, generalized phrase structure grammars (GPSGs) [34] have a CFG structure, where each nonterminal is augmented with a set of features. A unification mechanism is defined on these features through a set of unification constraints, which regulate the inheritance of the features in the parse tree. An efficient parsing algorithm for GPSG is proposed in [31]. History-based grammars (HBGs) [11] further address the use of context information while applying grammar rules. The rules are applied in a bottom-up fashion, and the context is represented by the neighboring words and nonterminals. Tree-adjoining grammars (TAGs) [87] are also context-dependent. They preserve the properties of CFGs, with augmented rules for generating parse trees and semantic interpretations according to context information. Head-driven phrase structure grammars (HPSGs) [84] are similar to GPSGs, except that the parsing is mainly driven by specific constituents like verbs and nouns called *heads*. Unification grammars associate a unification mechanism to rules based on theorem proving [103], which can be used to satisfy linguistic constraints such as number and gender agreements.

The grammar rules used in natural language understanding systems need to account for

both syntactic and semantic knowledge of the language. The basic approaches of designing the grammars can be classified into two types, namely the syntax-driven approach and semantic-driven approach, as indicated in [95, 113].

In classic natural language analysis for *written text*, a syntax-driven approach is usually taken. The words in the sentence are tagged according to their part of speech (POS) first. A syntactic grammar is used to parse the POS's into syntactic structure, such as noun phrase (NP), verb phrase (VP), prepositional phrase (PP), etc. A separate set of semantic rules are applied to obtain the semantic interpretation. One typical procedure is to construct one semantic rule for each syntactic rule in the grammar. It is assumed that the meaning of a phrase can be composed from the meaning of its constituents. For example, the meaning of the sentence S in the syntactic rule $S \Rightarrow NP\ VP$ is viewed as a function of the meaning of the noun phrase NP and the verb phrase VP. The semantic rule essentially defines a function that maps the meaning of NP and VP to the meaning of S. The complete meaning of a sentence is constructed by recursive application of the semantic functions, according to the syntactic analysis. Figure 4-3 illustrates a parse tree using a syntactic grammar.

The syntax-driven approach uses general syntactic categories in the grammar, and consequently has a relatively broad coverage. However, a syntactic grammar provides much less constraint than can be obtained by incorporating semantic information. Furthermore, this approach depends strongly on the syntactic soundness of input sentences, and a grammar is usually designed to handle well-formed written text. Therefore, a syntax-only approach may not be appropriate to use for natural language understanding in speech-based conversational interfaces. Many researchers have explored a semantic-driven approach instead. For example, the CMU air travel information service [109] uses RTNs to define key semantic phrases in the sentence, and the meaning representation is a flat structure of key-value pairs derived from the phrases. Syntactic structures are ignored. This is essentially a key-phrase spotting strategy, with emphasis on obtaining robust semantic information from the input sentences. Clearly, this might not be the optimal strategy when the correct meaning representations are dependent on a detailed syntactic analysis of linguistic constructs.

It is also possible to take advantage of both the syntax-driven approach and the semantic-driven approach by combining them. In the next section, we will introduce the TINA natural

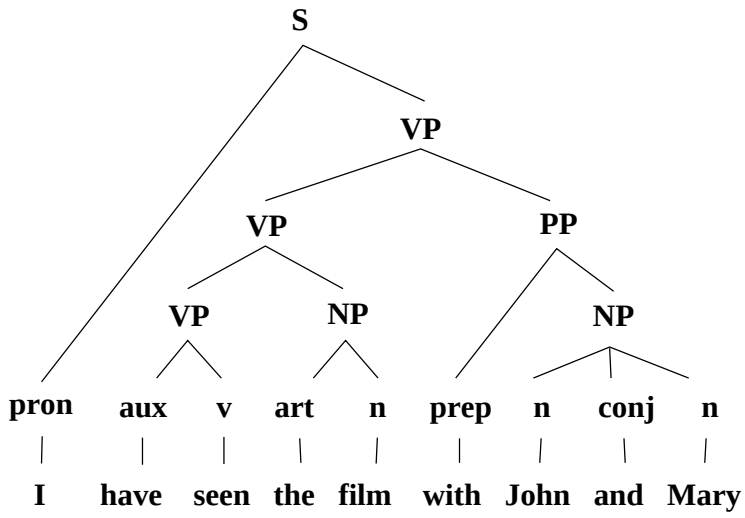


Figure 4-3: Sample parse tree of the sentence "I have seen the film with John and Mary", according to a syntactic grammar. The bottom layer above the words contains the POS tags, such as pronoun (pron), auxiliary (aux), verb (v), article (art), and preposition (prep). The parse nodes above the bottom layer are the nonterminals defined in the syntactic grammar, such as noun phrase (NP), verb phrase (VP), and sentence (S).

language system, which uses grammar rules that intermix syntax and semantics, and the meaning representation is derived directly from the resulting parse tree, without using separate semantic rules. Special efforts are also made in TINA to accommodate spontaneous speech inputs and erroneous recognition hypotheses.

4.3.3 TINA Natural Language System

Our study of supra-lexical linguistic modeling is based on TINA, which is a natural language understanding system for spoken language applications introduced in [94]. TINA is designed to perform linguistic analysis for speech recognition hypotheses, and to generate a semantic representation encoding the meaning of the utterance.

Like most natural language understanding systems, TINA uses a hierarchical CFG to describe the sentence structure. While the TINA framework supports grammars based on syntax alone, or on semantics alone, in our research in conversational systems we have been motivated to construct grammars that encode both syntactic and semantic structure in a

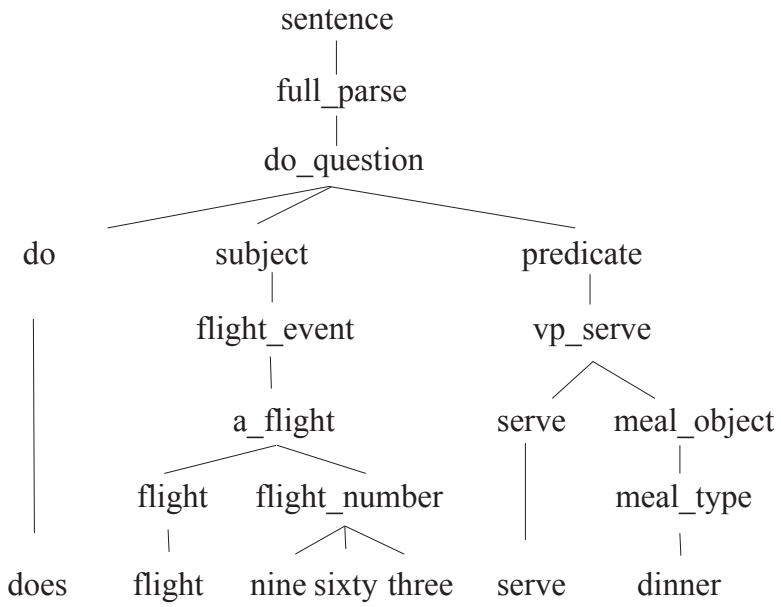


Figure 4-4: Sample TINA parse tree of the sentence "Does flight nine sixty three serve dinner". Syntax-oriented nodes, such as subject and predicate, are located at higher levels of the parse tree. Semantic-oriented nodes, such as "flight_number" and "meal_type", are located at the lower levels.

joint rule set. At the higher levels of the TINA parse tree, major syntactic constituents, such as subject, predicate, direct object etc., are explicitly represented through syntax-oriented grammar rules. The syntactic structures tend to be domain-independent, and capture general syntactic constraints of the language. At the lower parse tree levels, major semantic classes, such as "a_location", "a_flight" etc., are constructed according to semantic-oriented grammar rules. The semantic structures tend to be domain-dependent, and capture specific meaning interpretations in a specific application domain. Such a grammar is able to provide a more balanced solution between coverage and constraints. Furthermore, it also offers the additional convenience that no separate semantic rules are necessary for semantic analysis, and the semantic representation can be derived directly from the resulting parse tree. Figure 4-4 shows an example TINA parse tree.

In conversational systems, parse failure often occurs due to ungrammatical speech inputs and erroneous recognition hypotheses. In order to achieve a robust meaning analysis, a back-

off robust parsing strategy is used in TINA to deal with such ill-formed inputs. With robust parsing, a complete linguistic analysis is attempted first. If it fails, a partial parse will be performed to generate meaning from phrase fragments. The system will conduct key-word spotting based analysis in extreme cases where even phrase fragments are not available.

There are some other important features of the TINA system that are worth mentioning. First, a feature unification process is used to enforce some additional long-distance constraints, such as number (singular, plural) agreement, and verb mode (finite, present, past participle, etc.) constraints. This is achieved by augmenting the TINA vocabulary with a set of relevant features, then inheriting and unifying the compatible features according to the specified unification requirements. Second, a *trace* mechanism is used to handle long-distance movement phenomena. For example, the sentence "What city have you visited" shares the same canonical meaning representation structure as the sentence "Have you visited Boston". However, for the first sentence, "What city" has been moved to the beginning of the sentence in its surface form realization. The trace mechanism in TINA is able to move such constituents back to their underlying position, and generate canonical meaning representations. Details of these features can be found in [94] and [95].

4.4 Probability Augmentations for Natural Language Grammars

4.4.1 Overview

Like the probability augmentations for sub-lexical grammars discussed in section 2.4, probability frameworks are also used to augment natural language grammars. There are two basic reasons for introducing statistics in the natural language component of a speech understanding system:

- The necessity of assessing speech recognition hypotheses

  The natural language component in a speech understanding system needs to evaluate different speech recognition hypotheses, and choose the best candidate that complies with the *a priori* linguistic knowledge of a particular language. Therefore, it is neces-

sary to estimate the likelihood that each hypothesis would occur in the language given the syntactic and semantic constraints. Furthermore, the natural language grammar can be ambiguous, so that more than one legitimate parse tree is associated with one sentence. It would be convenient to have a probability estimation for each parse tree, and the most likely result can be used for meaning generation.

- The necessity of introducing stronger constraints

  As was indicated before, it is necessary to balance the coverage and constraints for natural language grammars. On the one hand, a probability framework for a natural language grammar can help impose additional constraints without rewriting of the grammar and sacrificing the coverage. On the other hand, the structure analysis provided by the natural language grammars can also help establish a more constraining probability framework. For example, in the TINA parse tree example given in Figure 4-4, the word "serve" can often be much more effectively predicted by the left category "a_flight" than by immediately preceding words used in an n-gram model (in the example, "sixty three").

Many researchers have investigated possible probability frameworks for language processing. For example, probabilistic Tree Adjoining Grammar (TAG) [87], probabilistic LR parsing [15], probabilistic link grammar [55] etc. In this section, we focus on two types of probability frameworks. We first outline the typical probabilistic augmentation approach using Stochastic Context-free Grammars, and then discuss structurally-based context-dependent hierarchical probability models used with CFGs.

4.4.2 Stochastic Context-free Grammars

One basic probability augmentation used with natural language grammars is called the Stochastic Context-free Grammar (SCFG). In a SCFG, a probability is associated with each rule expansion. The probability of a particular parse tree is defined by multiplying the probabilities for each individual rule used in the derivation, assuming the rule applications are independent of one another. A probability distribution is associated with the set of all the parse trees generated by the grammar.

An iterative training approach called the *inside-outside* algorithm [8] is used to train a SCFG. This is a variant of the EM algorithm [28], widely used for parameter estimation of statistical models. The basic training procedure can be described as follows. We start with some initial rule probability estimation, and a large training corpus is parsed according to the CFG. In each iteration, the probabilities are boosted for the rules that participate in successful parses, and lowered for those that do not. The procedure terminates when the change of probabilities falls below a certain threshold.

Note that the sub-lexical rule-production probability model discussed in section 2.4.1 is essentially a SCFG. The major difference is that the sub-lexical grammars are highly regulated and usually less ambiguous than the natural language grammars.

SCFGs are able to provide a reasonable estimation of parse tree probabilities; however, as in the sub-lexical case, they cannot model context dependencies across rule-boundaries. In some situations, categorical contexts in a parse tree beyond the current derivation are helpful for imposing stronger probability constraints, particularly the semantic-oriented categorical contexts in an intermixed grammar. The semantic information carried by such semantic categories can be a strong predictor of the words to follow. For example, the lower level parse tree nodes in the TINA grammar can be good predictive contexts. Next, we will discuss structurally-based hierarchical probability models that are able to take this consideration into account.

4.4.3 Context-dependent Hierarchical Probability Model

As was pointed out in section 2.4.2, the central issue in designing a context-dependent probability model with CFGs is to choose the conditional context. In our study of hierarchical sub-lexical linguistic modeling, the sub-lexical grammars were highly regulated. The non-terminals in the grammars are organized into layered groups, and no layer is skipped in a successful parse. This allows us to use a uniform tabulated representation of the parse tree, as illustrated in Table 2-1. With this uniform parse representation, it is possible to use full columns in the tabulated form as conditional contexts for the phone layer. We have also demonstrated in section 3.7 that, with the amount of sub-lexical training data available, the use of full column contexts does not result in a serious sparse training data problem and,

at the same time, it can provide rich context information across the sub-lexical linguistic hierarchy.

At the supra-lexical level, however, the natural language grammars used in language understanding are usually much more complex. There is no uniform layered parse tree structure, and some parse tree branches may be much deeper than others. If the entire path from the root node to the leaf node was chosen as the conditional context, as we did in sub-lexical modeling, insufficient training data would be a serious concern. The goal here is to provide probabilistic context-dependent support, keeping the model size manageable in order to conduct robust training.

Our study of applying context-dependent probabilities with CFGs is based on the TINA probability model, which is an important component of the TINA natural language understanding system. It provides hierarchical context-dependent probabilistic constraints in terms of a trainable grammar. According to the TINA probability model, the probability of a parse tree is constructed from the probability of each parse tree node conditioned on its parent and left-sibling. There are two types of probabilities in TINA: the rule-start probability, and the rule-internal probability. The details are given below:

- Rule-start Probability

  The rule-start probability is specified for a rule-start node given its parent and its left-sibling across adjacent rules in the parse tree. For example, in the TINA parse tree given in Figure 4-4, the non-terminal node "vp_serve" is a rule-start node in the derivation "predicate $\Rightarrow$ vp_serve", and its probability is defined by the rule-start probability $P(\text{vp\_serve} \mid \text{predicate, flight\_event})$, which is the probability of the non-terminal node "vp_serve" conditioned on its parent "predicate", and its left sibling "flight_event" beyond the current rule boundary. Such context-dependent rule-start probabilities are mainly used to capture the context-dependencies across adjacent rules.

- Rule-internal Probability

  The rule-internal probability is specified for a rule-internal node given its parent and its left-sibling within the grammar rule. For example, the non-terminal node

"meal_object" in Figure 4-4 is a rule-internal node in the derivation "vp_serve ⇒ serve meal_object", and its probability is defined by the rule-internal probability $P(\text{meal\_object} \mid \text{vp\_serve, serve})$, which is the probabilities of the non-terminal node "meal_object" conditioned on its parent "vp_serve" and its left sibling "serve" within the rule.

Given the rule-start and rule-internal probabilities, the probability of a parse tree is defined by the product of the conditional parse tree node probabilities. Depending on the position of each node, rule-start or rule-internal probabilities are applied. We also experimented with the *generic* rule-start probability, which refers to the probability of a rule-start node conditioned on its parent and a generic rule start marker. Note that the rule production probabilities in a SCFG can be approximated by multiplying the generic rule-start probability and the rule-internal probabilities. Such approximation may be desirable with limited training data, because it allows probability sharing of parse tree nodes with the same parent and left sibling context. With the context-dependent rule-start probability, it provides as well probability support for context-dependent rule derivations, which is beyond the standard SCFG formalism.

As was discussed in section 4.2.2, structured language models used in speech recognition also provide a context-dependent probability framework with formal grammars. Despite the fact that the grammars used for structured language models and natural language understanding are typically quite different, the context-dependent probability frameworks share some fundamental similarities. They all associate a probability distribution over possible parses defined by the grammar, and the parse tree probability is defined with the consideration of parse tree context information. In the structured language model defined by equation 4.6, the partial parse trees are used as conditional contexts, and special efforts are made to cluster such contexts to alleviate the sparse data problem. In the TINA context-dependent probability model, current parents and left siblings are used as conditional contexts. The context selection is a key issue in designing an effective context-dependent probability model. In the next section, we will present our study on automatic left context selection for the hierarchical context-dependent models.

4.5 Automatic Context Selection for Hierarchical Probability Models

In context-dependent hierarchical probability models, selecting effective context is an important factor for model development. Since the hierarchical grammar used at the supra-lexical level is much more complex than at the sub-lexical level, context clustering is necessary for robust training. Some systems choose representative parse tree nodes to consolidate parse tree contexts, such as the *exposed heads* [16] of partial parse trees. In the TINA probability model, the parent and left sibling nodes are used as conditional contexts.

In this section, we study the selection of conditional context for the rule-start probabilities based on the TINA probability model. An automatic procedure is proposed to identify effective left context nodes. The experiments are conducted in the JUPITER weather information domain, with the JUPITER natural language grammar. A development set is used to adjust the pruning thresholds. The same full test set as was used in the sub-lexical modeling studies is used for perplexity testing. The experimental results and conclusions are also presented.

4.5.1 Obtaining Effective Left Context

As was discussed in section 4.4.3, the TINA context-dependent probability model uses the left sibling as the left context for rule-start probabilities. This may not be the optimal way of laying out the probability model, since the left sibling may not always have the strongest predictivity for the next rule-start word. In TINA's grammar, certain categories, especially some semantic-oriented categories, are better context representatives than other categories. Therefore, it may be advantageous to be able to identify the promising categories and use them as the conditional contexts.

In this study, we have proposed the following procedures to automatically select effective left context nodes. First, rule-start statistics are collected for every left context node, including the left sibling and other immediate left parse tree nodes, such as left parent, left child, etc. The generic rule-start statistics mentioned in section 4.4.3 are also collected, which do not carry context-dependent information across rule boundaries. Then, *symmet-*

ric divergence based on KL-distance [54] between a context-dependent rule-start probability distribution and the corresponding generic rule-start probability distribution is calculated. KL-distance is a measurement of the directional distance from one probability mass distribution to another, and is known as *relative entropy* in information theory. The symmetric divergence is the sum of two asymmetric directional KL-distance measurements, which measures the similarity between the two probability distributions. When a context node results in a large symmetric divergence from the generic rule-start probability distribution, it is a good indication the such a node may carry more effective context information. Next, the rule-start probability models are pruned to obtain a more robust model. Pruning can also be used to reduce the model size. Two pruning strategies are explored: the first one prunes the context-dependent rule-start probability models with insufficient training observations, and the second one prunes those with probability distributions similar to the generic rule-start probability distribution according to the KL-divergence. Finally, the left context node with the highest KL-divergence is chosen as the conditional context.

4.5.2 Pruning Context-dependent Rule-start Probability Models

The first pruning experiment is to eliminate ill-trained probabilities due to insufficient training observations. Although only the parent node and one left context node are used as conditional contexts, it has been observed that there are still a lot of rule-start probability models that do not have enough training data. Therefore, it is necessary to prune away such ill-trained probabilities. In this experiment, the pruning criteria are defined from the average observation count for estimating a conditional probability distribution. Probability models with average observation counts less than the pruning threshold are pruned. Figure 4-5 shows the model perplexity measured on the JUPITER development set as a function of the average count pruning threshold. The number of model parameters is also shown.

As we can see, with the increase of the average count pruning threshold from 1.0 to 2.0, the model achieves a lower perplexity with smaller size. This is desirable, and the ill-trained models are effectively eliminated. With further increase of the pruning threshold, more aggressive pruning results in further reduction in model size, and the perplexity starts to increase. We choose to set the average count pruning threshold to 3.0, which gives a

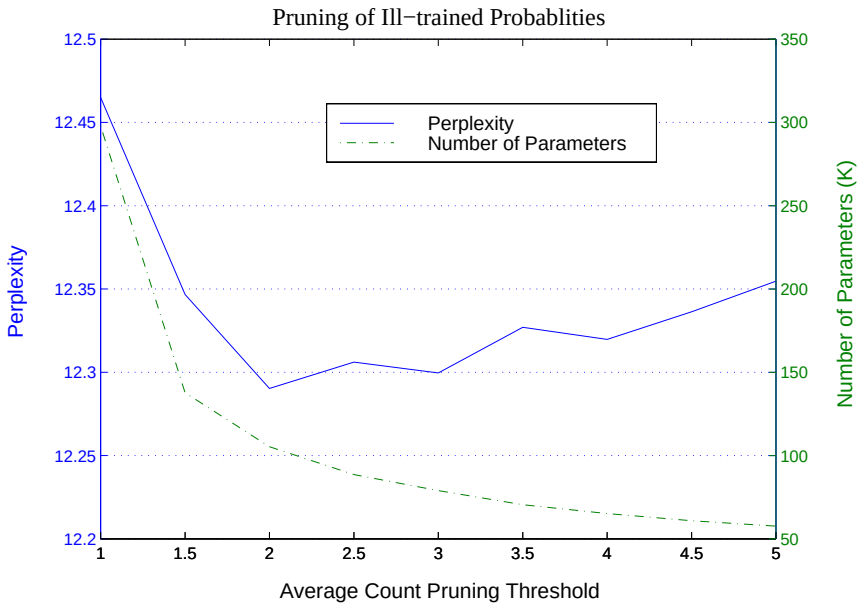


Figure 4-5: Pruning of ill-trained probabilities on the development set. Model trained on the full training set. Both perplexity (left y-axis) and model size (right y-axis) are shown as a function of the average count pruning threshold.

relatively small model with reasonably low perplexity.

Another possible way to reduce the model size is to prune away the context-dependent models with probability distributions similar to the generic rule-start probability distribution. The divergence based on KL-distance can be used to measure such similarity. Figure 4-6 shows the results of applying such a strategy after pruning the ill-trained probabilities. We can see from the figure that this KL-distance based pruning is also able to reduce the number of parameters, resulting in a compact model. However, since the pruned probabilities are relatively well-trained, the perplexity increases monotonically as more aggressive pruning is applied. In the following automatic context selection experiments, we choose to keep the relatively well-trained probabilities, and such KL-distance based pruning is not used.

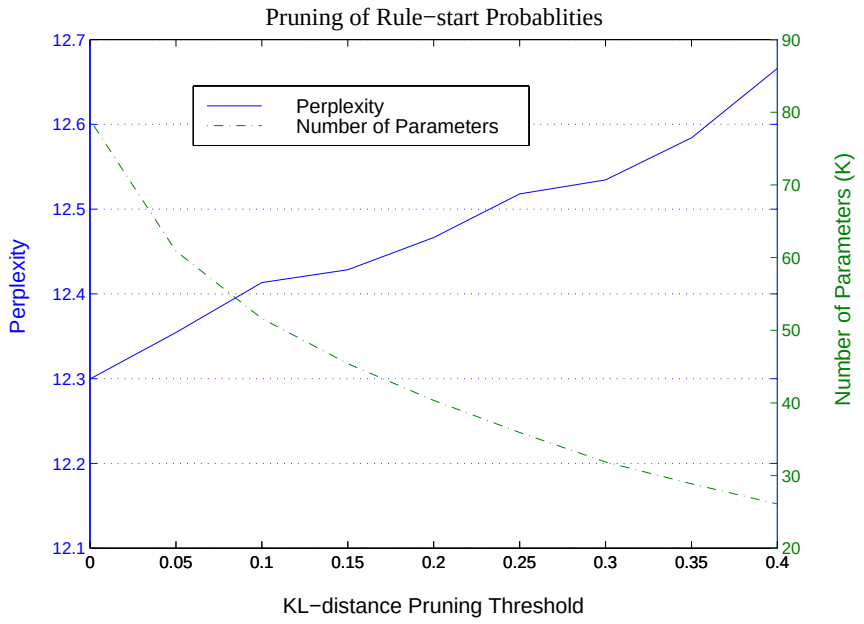


Figure 4-6: Pruning of rule-start probabilities on the development set. Model trained on the full training set. Both perplexity (left y-axis) and model size (right y-axis) are shown as a function of the KL-distance pruning threshold.

4.5.3 Perplexity Results of Automatic Context Selection

After pruning the ill-trained models with insufficient training observations, the left context node corresponding to the highest KL-divergence from the generic rule-start probability distribution is selected as the conditional context for the current rule-start node. Pruning ill-trained models is important here, because ill-trained models can lead to large KL-divergence, which does not necessarily imply more effective context-dependent probabilities. Compared to the original model, where the left sibling is always designated as the conditional context, this approach allows the flexibility of choosing the rule-start probability model with the most divergent conditional probability distribution, which may indicate a stronger predictability for the next rule-starting node.

We have compared three configurations for this context selection experiment. The first one uses the generic rule-start probability only, without left context information. The second one uses the original TINA left sibling context. The third one uses the automatic

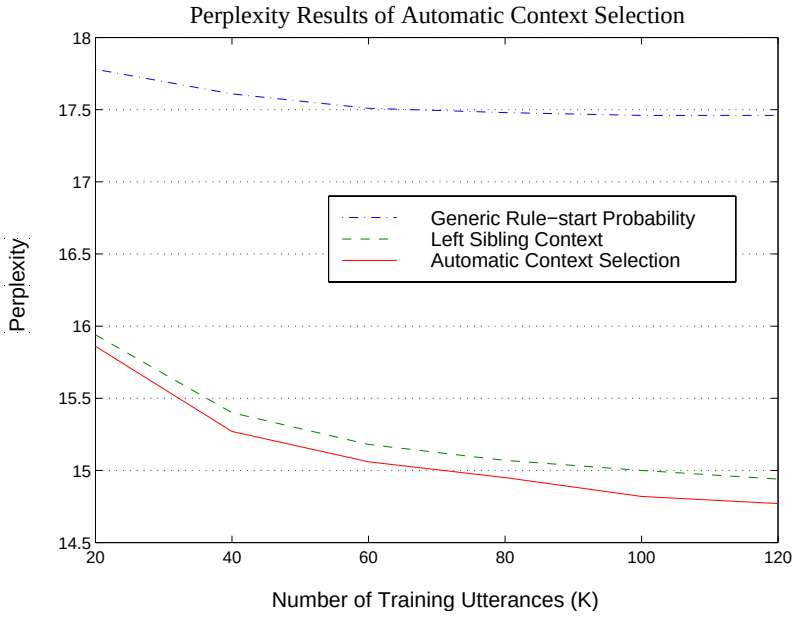


Figure 4-7: Perplexity results for automatic context selection. The perplexity is shown as a function of the amount of training data used.

left context node selection approach described above. Figure 4-7 shows the perplexity results on the full test set as a function of the amount of training data used. From the results we can see that using context-dependent probability models has achieved a much lower perplexity than using the generic rule-start probability model. This suggests that the additional context-dependent rule-start information is effective in helping constrain the hierarchical probability model. The use of automatic context selection leads to further improvements, and such improvements are consistent with different amounts of training data. It can also be observed that the automatic context selection results in relatively more perplexity reduction with increasing training data size. One possible reason is that the probability and KL-divergence estimations tend to be more robust with more training data.

We further studied the statistics on the left context nodes chosen by the automatic context selection procedure. Figure 4-8 shows a histogram of the frequency of left context nodes chosen at different levels relative to the left sibling node. Cumulative hit rates from the top parse tree level are also displayed. We see that, with the particular JUPITER grammar we use, more than 80% of the left context nodes chosen are at the left sibling level or above.

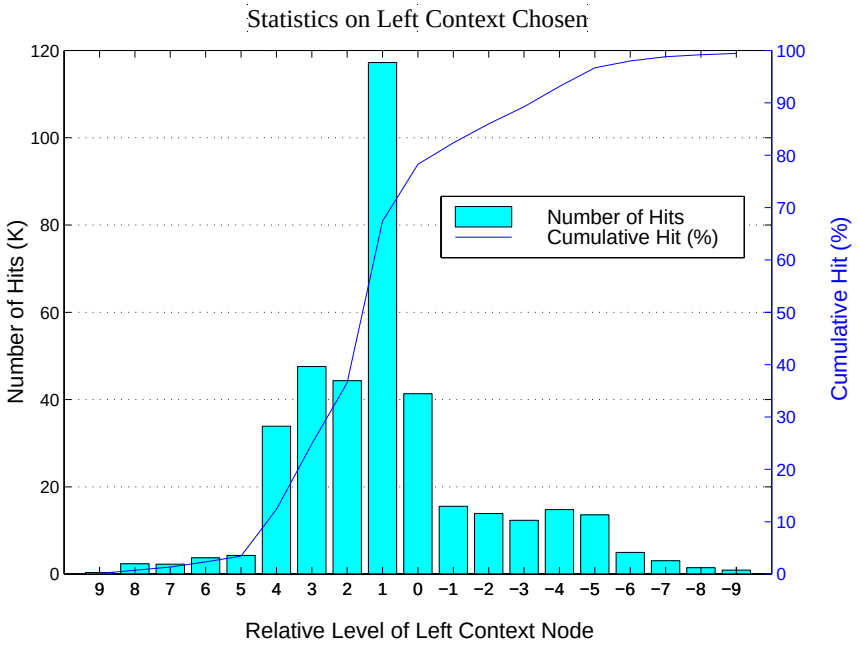


Figure 4-8: Statistics on the left context nodes chosen. The number of hits for each relative level of the left context node is shown (relative level 0 is the left sibling, relative level 1 is the left parent, relative level -1 is the left child, etc.). The cumulative hit percentage is also displayed.

We also notice from the histogram that the left parent seems to be the most effective context node in providing context-dependent constraints. These observations suggest some possible simplifications to the automatic context selection procedure. One motivation for using a simplified approach is to reduce the run-time computation for context-selection. As was discussed in section 4.5.1, the KL-divergence measurements are computed for all possible contexts, which can be expensive. To simplify the context selection algorithm, we can limit the left context selection within the levels above left sibling node, or simply use the left parent as the conditional context. Figure 4-9 shows the perplexity results using the latter approach. We see that this does provide a better probability model than using the left sibling context. It is not quite as good as the full-range automatic context selection model. The perplexity improvements are moderate, however, which implies that it is a reasonably good choice to use the left sibling context in the original TINA model.

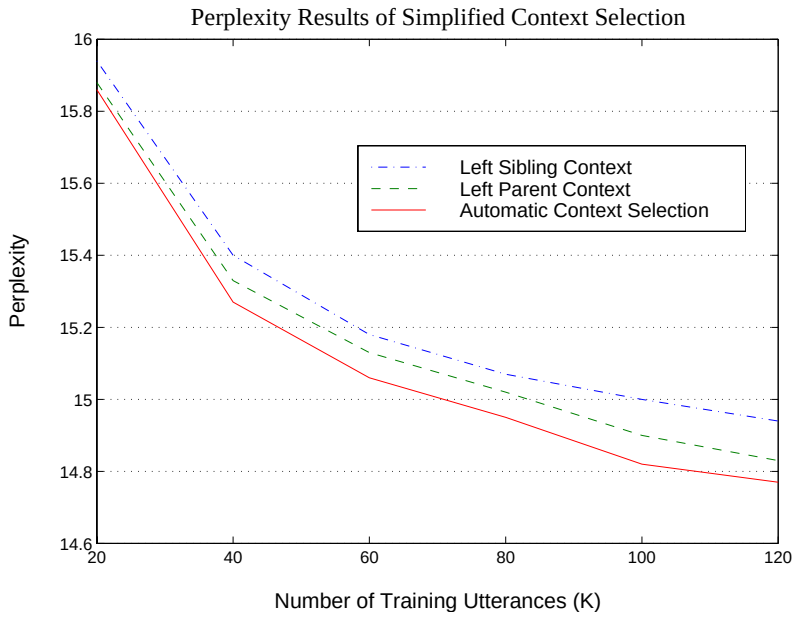


Figure 4-9: Perplexity results for the simplified context selection approach. The perplexity is shown as a function of the amount of training data used.

4.6 Summary

In this chapter, we discussed the use of supra-lexical linguistic knowledge in both speech recognition and language understanding components, and proposed an approach to automatically select the conditional context in the context-dependent hierarchical probability model.

The widely used supra-lexical linguistic model in speech recognition is the n-gram language model, because it can provide local probability constraints with the flexibility of accepting arbitrary word sequences, and is simple to use. Several n-gram variants are proposed to improve the basic word n-gram model, for example, class n-gram models are introduced to reduce the number of model parameters, which can help obtain a relatively robust model with limited training data. Predictive clustering further introduces class context-dependency while predicting the next word given the class it belongs to. Beyond the n-gram formalism, structured language models in speech recognition have been explored by many researchers. They address sentence structure constituents and long-distance con-

straints with the use of a formal grammar, typically a CFG.

The natural language understanding component also makes heavy use of supra-lexical linguistic knowledge. Typically, natural language grammar rules are used to formalize the syntactic and semantic knowledge, and a probability model is imposed on the grammar. The probability model is used to provide additional constraints, and offers a quantitative assessment of how well the input word sequence complies with the *a priori* linguistic knowledge. The basic approach to augment natural language CFGs is to use SCFGs. However, SCFGs are not able to model context dependency beyond the rule boundaries. Context-dependent probability models can be used to account for categorical contexts in a parse tree beyond the current derivation; thus, stronger probability constraints can be imposed. The central issue of designing a context-dependent probability model to be used in conjunction with CFGs is to choose the conditional context.

We have proposed an approach to automatically select the effective conditional context used in hierarchical context-dependent linguistic modeling, based on the TINA probability model. It has been shown that the context-dependent rule-start probability model is clearly more effective than the generic rule-start probability model. The automatic context selection approach is able to provide further improvements. After studying the statistics on the left context chosen, we have explored one simplified scheme, where the left parent is used as the conditional context instead of the left sibling. Experimental results show that using the left parent as the conditional context results in a model with lower perplexity. The perplexity improvement is moderate, which suggests that the original left sibling context is also effective.

Chapter 5

Integration of Hierarchical Supra-lexical Models with Speech Recognition

5.1 Overview

In this chapter, we study the integration of hierarchical supra-lexical linguistic models with speech recognition. As we have seen in Chapter 2 and Chapter 4, hierarchical linguistic knowledge is available at both sub-lexical and supra-lexical levels. At the sub-lexical level, the linguistic hierarchy is presented by sub-lexical linguistic knowledge such as morphology, syllabification, phonemics, phonetics, etc. Such a hierarchy is typically formalized by sub-lexical grammars. At the supra-lexical level, formal hierarchical grammars are also used in structured language models to address limitations of basic n-gram models in speech recognition. Such grammars are usually syntax-oriented, and help impose syntactic structural constraints. Further hierarchical linguistic analysis in syntax and semantics is often performed in the natural language understanding component.

There are some fundamental similarities between hierarchical sub-lexical and supra-lexical linguistic models. They are all based on hierarchical grammars, usually in the form of CFGs. The grammars are used to provide sub-lexical and supra-lexical structural

constraints of the language. Moreover, such grammars can be augmented with context-dependent probability models at both levels beyond the standard CFG formalism. In addition to giving an estimation of the likelihood that some linguistic unit sequence may occur in the language, such context-dependent probability augmentation can provide context-dependent probability constraints superimposed on linguistic structure. We have seen in Chapter 3 that modeling context-dependency is essential at the phonetics layer of the sub-lexical hierarchy. In Chapter 4, experimental results also show that at the supra-lexical level, using context-dependent rule-start probabilities can achieve lower perplexities than generic rule-start probabilities. In Chapter 3, we have developed an FST-based framework to integrate the context-dependent hierarchical sub-lexical model into speech recognition. Given the similarities mentioned above, it can be advantageous to use a unified FST-based framework to incorporate supra-lexical hierarchical linguistic knowledge into speech recognition as well.

A natural extension to the approach of separately incorporating sub-lexical or supra-lexical hierarchical linguistic constraints within the FST-based framework is to establish a unified speech recognition search space, which encodes the linguistic constraints at both levels. This extension would result in a tightly coupled system, with linguistic knowledge applied at early stages of the recognition search. One major concern for such an approach, however, comes from the complexity of the natural language grammars. Unlike sub-lexical grammars, which usually produce a uniform parse tree structure, supra-lexical grammars typically do not have such regularity. Some form of simplification is necessary to facilitate the construction of FST-based models at the supra-lexical level. We will see in this chapter that the complexity of FSTs becomes an important problem of integrating hierarchical supra-lexical linguistic knowledge into speech recognition. It is also one major obstacle of establishing a complete system including both sub-lexical and supra-lexical linguistic knowledge.

Another factor concerning supra-lexical linguistic modeling is that supra-lexical linguistic knowledge is used in both speech recognition and natural language understanding. In a typical speech understanding system, these two components are integrated with an N-best list or a word graph, and the natural language understanding component acts as a

post-processor of the recognition hypotheses. This is basically a feed-forward system, with no feedback from natural language understanding to guide the original speech recognition search. Using a unified framework to incorporate supra-lexical linguistic knowledge in speech recognition can result in a tightly coupled interface, which offers early integration of linguistic constraints provided by natural language understanding.

As was pointed out in section 4.1, although supra-lexical linguistic knowledge is used by both speech recognition and natural language understanding in a speech understanding system, the constraints applied may not be consistent. The development strategies for the two components are usually different. Most speech recognition systems use corpus-based n-gram models, due to their simplicity, broad coverage, and satisfactory performance in most cases. Most natural language understanding systems, however, use linguistically motivated rule-based approaches, because linguistic analysis is necessary for meaning generation. Since speech recognition hypotheses need to be processed by the natural language understanding component in a speech understanding system, it can be advantageous to impose speech recognition constraints more consistently with natural language understanding constraints.

In summary, our main objective in integrating supra-lexical linguistic knowledge into speech recognition is towards establishing a unified FST-based framework that imposes supra-lexical linguistic constraints consistent with natural language understanding. In the following sections, we first introduce the FST-based speech recognition framework with some typical supra-lexical linguistic models. Then, a phrase-level shallow parsing approach is proposed, which simplifies the parse tree structure and facilitates the construction of context-dependent probability models in a unified FST framework. The shallow parsing grammars are directly derived from the grammars for natural language understanding. Next, we discuss the construction of the shallow-parsing-based supra-lexical linguistic model using layered FSTs similar to what has been used at the sub-lexical level. Our supra-lexical level speech recognition experiments are conducted in the JUPITER domain, and the experimental results and conclusions are presented. Finally, we summarize this chapter and discuss the advantages and disadvantages of such an FST-based supra-lexical linguistic modeling approach.

5.2 Supra-lexical Linguistic Modeling Framework Using FSTs

Analogous to the FST-based sub-lexical linguistic modeling approaches discussed in section 2.5, FSTs can also be used at the supra-lexical level. Many supra-lexical linguistic models can be represented directly by FSTs, such as word graphs and basic n-gram language models. Some of the n-gram model variants, such as the class *n-gram* model, can also be represented in the FST framework using RTNs. Moreover, RTNs can be used to describe hierarchical natural language grammars as well.

Some of the advantages of using FSTs at the supra-lexical level are similar to those at the sub-lexical level. An FST framework provides a parsimonious representation of supra-lexical linguistic knowledge, which can be seamlessly integrated with other knowledge sources in a FST-based speech recognizer. The recognition search is performed in a uniform search space defined by the composition of FSTs, which allows consistent and global optimization criteria to be applied.

Using FSTs at the supra-lexical level may also provide some additional benefits. For example, FSTs can be used to impose hierarchical constraints derived from natural language understanding, which offers a tightly coupled interface capable of giving feedback from natural language understanding to speech recognition at an early stage. However, since the natural language grammars are much more complex than sub-lexical grammars, such a tightly integrated system may result in a significantly larger search space because of the increase in FST complexity. Therefore, the use of such a system can be highly restricted by the computation requirements.

In this section, we reiterate briefly the top-level recognition architecture based on FSTs, and address some potential problems in applying supra-lexical linguistic constraints using FSTs. Then, we discuss the details of some FST-based supra-lexical linguistic models, including FST-based n-gram models and some of their variants, and FST-based hierarchical models.

5.2.1 FST-based Speech Recognition with Supra-lexical Linguistic Models

As was discussed in section 2.5, the speech recognition search space can be defined through the composition of FSTs representing various acoustic and linguistic knowledge. For example, a typical segment-based speech recognizer can be represented by the FST composition given in expression 2.5:

$$S \circ A \circ C \circ P \circ L \circ G \tag{5.1}$$

where the composition $S \circ A \circ C \circ P \circ L$ encodes acoustic and linguistic knowledge below the word level (see details in section 2.5), and G encodes the supra-lexical linguistic knowledge. FST G can be a simple word graph, or a more complex supra-lexical model that represents probability and/or structural constraints of the language.

In principle, the FSTs can be pre-composed and optimized by standard optimization procedures such as ϵ-removal, determinization, and minimization [69, 72, 70]. However, such pre-composition and optimization become increasingly difficult as the complexity of each FST increases. This is a more serious problem at the supra-lexical level. With a large vocabulary size, even basic tri-gram model FSTs can hardly be pre-composed and optimized [30]. One obvious solution is to compose such FSTs on-the-fly during the recognition search. However, this would lose the advantage of having an optimized search space. Some researchers have addressed such a problem of constructing FST-based large vocabulary n-gram models using an *incremental* modeling approach [30]. In this case, the original n-gram language model FST G is factored into two FSTs. The first FST is composed statically with the sub-lexical acoustic and linguistic FSTs, and optimized in advance. The second FST is composed on-the-fly during recognition search. Such an approach is able to include part of the supra-lexical linguistic model FST into early optimization, while keeping the complete supra-lexical model through incremental dynamic composition. For hierarchical supra-lexical models that incorporate structural constraints, simplifications either in the grammars or in the probability models may be necessary to reduce the complexity of the FSTs.

5.2.2 FST-based n-gram Language Models and Their Variants

Basic n-gram language models can be directly represented by FSTs. The overall FST structure is the same as the sub-lexical level phoneme n-gram model illustrated in Figure 3-4. The n-gram contexts are remembered by FST context states, and the n-gram conditional probabilities are encoded by weighted transitions between the context states. Weighted ϵ transitions are used to encode the back-off weights for model smoothing, which was explained in equation 4.2. The major difference between supra-lexical and sub-lexical n-gram FSTs is that, at the supra-lexical level, the basic linguistic units are typically words, and the number of unique words in a large vocabulary system can be much greater than the number of basic linguistic units used at the sub-lexical level, such as phonemes or phones. Therefore, the supra-lexical language model FST can be much larger, which makes it difficult to optimize.

We have seen that class n-gram language models can be used to reduce the number of model parameters, resulting in a smaller language model. The class rules can be viewed as a trivial CFG, with class labels at the left hand side of the grammar rules, and individual words at the right hand side. Therefore, it is possible to represent the class n-gram model by RTNs, which can be compiled into a regular FST in this trivial CFG case. The top-level network in the RTN represents the class level n-gram model. It shares the same structure as a regular word n-gram FST, except that some transitions are labeled with class names. Further class expansions are represented by weighted individual sub-networks, which encode the word probabilities within each class. Figure 5-1 illustrates the FST structure for a class bi-gram model.

Note that, with RTNs, it is not straightforward to model the context-dependency across adjacent classes for word realizations, which is applied in predictive clustering discussed in section 4.2.1. The context-dependent probabilities for words can not be directly encoded by weighted transitions within each sub-network, because the probability of a word is conditioned not only on the class it belongs to, but also the previous classes. It is possible to build such context-dependent probability models using the layered FST approach proposed in section 3.4, where the CFGs are represented by RTNs, and context-dependent probabilities are captured using separate FSTs.

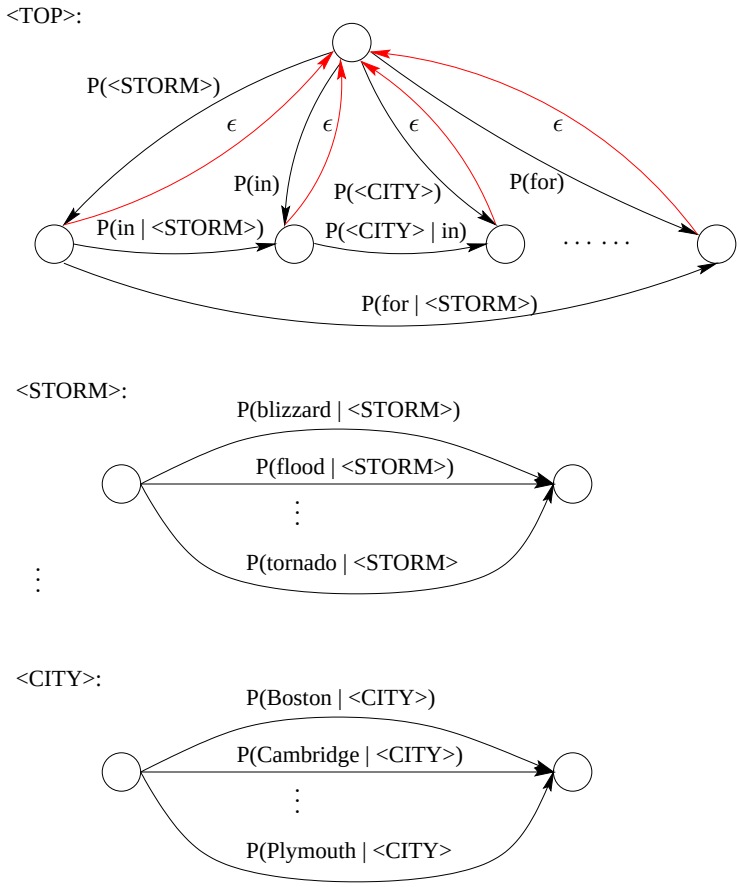


Figure 5-1: FST topology for a class bi-gram model. The top-level network has the same structure as a regular word bi-gram model, except that some transitions are labeled with classes. Other sub-networks represent the class expansions, and are weighted according to the probabilities of each word within the class.

5.2.3 FST-based Hierarchical Supra-lexical Linguistic Models

In Chapter 4, we have seen that supra-lexical level sentence structure can be described by hierarchical formal grammars, usually in the form of CFGs. In principle, the CFGs used to describe sentence structure can be represented with RTNs. Rule production probabilities used in SCFGs can also be encoded directly by the transition weights within the sub-networks. One important problem, however, is that, unlike the grammars used for hierarchical sub-lexical linguistic modeling, which are often highly regulated, the natural language grammars are much more complex. Furthermore, the underlying language defined by the natural language grammar can be context-free, and the corresponding RTNs may

not be pre-compiled into an FST using a finite number of states. It is therefore necessary to apply some simplifications in order to obtain a manageable FST-based model.

One way to apply such a simplification is to approximate context-free languages defined by CFGs with regular languages, which can then be represented by FSTs. The regular approximation can either be a superset or a subset of the original context-free language. Possible approaches include superset approximation through pushdown automata, subset approximation by transforming the grammar [79], etc. It is difficult, however, to approximate CFG-based hierarchical models augmented with context-dependent probabilities, such as the structured language model used in speech recognition and the context-dependent hierarchical model used in natural language understanding. Special efforts are needed to encode the context information using the FST states.

Another approach is to use CFGs to describe only the important structural constituents in the sentence. The local constraints between such structural constituents and filler words can be modeled by a regular n-gram model. The simplified CFGs are represented by RTNs, and the n-gram model can be folded into a top-level weighted network. The overall FST structure is similar to the FST-based class n-gram model illustrated in Figure 5-1, except that the trivial word class sub-networks are replaced by sub-networks that describe the sentence structural constituents according to the CFGs. This is essentially a hybrid model combining n-gram models and hierarchical models as discussed in section 4.2.2. Such a hybrid model also offers enhanced coverage with relatively tight structural constraints. Similar to the class n-gram case, the probabilities are typically conditioned within each sub-network of the RTNs. Special considerations are needed to incorporate context-dependent probability models across the rule boundaries. In section 5.4, we propose an FST-based layered approach, which is similar to what has been used at the sub-lexical level. It can be applied to construct context-dependent probability models with phrase level shallow parsing, which produces a simplified two-layer parse tree structure.

5.3 Integration of Language Constraints Using Shallow Parsing

In this section, we propose a shallow parsing approach for integrating supra-lexical linguistic constraints into speech recognition. The term "shallow parsing" is typically used as a generic term for analyses that are less complete than the output from a conventional natural language parser. A shallow analyzer may identify some phrasal constituents of the sentence, without presenting a complete sentence structure. As was pointed out in section 5.1, it can be advantageous to have supra-lexical speech recognition constraints consistent with natural language understanding. In this thesis, our shallow parsing grammar is derived directly from the full-fledged TINA natural language grammar, and the same phrase structure constraints are applied to speech recognition and natural language understanding.

In text-based natural language processing, shallow parsing approaches typically identify the syntactic constituents of a sentence. One example is the stochastic tagging procedure designed in [22], which can locate simple noun phrases according to syntactic noun phrase rules. Another example is the *Fidditch* parser presented in [43], which can be used to produce annotated syntactic phrase structure trees. However, in speech understanding systems, the grammars used can be semantic-driven, or intermixed with syntax-based and semantic-based rules, as was discussed in section 4.3.2. For example, the TINA natural language grammar produces a parse tree with syntax-oriented categories at higher levels, and semantic-oriented categories at lower levels. Since correct identification of meaningful phrases during speech recognition may benefit subsequent meaning analysis in a speech understanding system, the shallow parsing grammars used in our study are constructed to describe phrases with specific semantic labels, such as "month_name", "daytime", "disaster", etc. This is different from the classic shallow parsing approach used in text-based natural language processing.

5.3.1 Motivations of Shallow Parsing and Related Research

Our motivations for adopting a shallow parsing approach to supra-lexical linguistic modeling are summarized as follows. First, shallow parsing can help balance sufficient generality and tight constraints. Shallow parsing does not attempt to perform complete linguistic

analysis on the input sentence. Instead, it tries to identify phrasal constituents according to phrase level grammars. In fact, in the natural language understanding component of a speech understanding system, much effort has been devoted to analyzing sentences that do not fully comply with a pre-defined grammar. For example, about 14% of the utterances in a JUPITER test set can not be completely parsed using the TINA natural language grammar. In these cases, the TINA robust parsing strategy mentioned in section 4.3.3 tries to perform meaning analysis from phrase fragments. The shallow parsing approach in supra-lexical linguistic modeling shares a similar strategy, which provides tight phrase-level long-distance constraints, as well as broad coverage allowing arbitrary phrase and word connections at the top level. Second, shallow parsing facilitates the use of context-dependent probability models in an FST framework. As was shown in section 4.5, at the supra-lexical level, context-dependent hierarchical models have a lower perplexity than context-independent models. The context-dependent probability augmentation to CFGs can provide additional context-dependent probability constraints, which can be helpful in speech recognition. In section 5.3.2, we will show that shallow parsing can be used to produce a regulated two-layer parse tree. Such a regulated parse tree representation facilitates applying context-dependent probability models in a unified FST-based framework. The details of context-dependent supra-lexical linguistic modeling using shallow parsing and FSTs will be presented in section 5.4. Third, shallow parsing is able to simplify the full natural language grammar. We have mentioned that an FST-based hierarchical supra-lexical linguistic model consistent with natural language understanding can result in a tightly coupled system with early feedback. However, the integrated search space can be significantly larger with complex natural language grammars. Shallow parsing is based on a set of simplified phrase level grammar rules, which are easier to manage. We will see in section 5.4 that, even with a simplified shallow parsing grammar, the use of such a hierarchical supra-lexical linguistic model is still restricted by the complexity of the resulting FSTs.

Many researchers have explored the integration of structured linguistic constraints using a method similar to shallow parsing. For example, Ward [110] proposed a method of integrating semantic constraints into the recognition search using phrase level RTNs. The phrase class n-gram model [65] discussed in section 4.2.2 can also be viewed as a shallow

parsing strategy. Another example is the hierarchical phrase language model studied by Wang [108]. In this approach, weighted RTNs are used to describe hierarchical phrase structure. Similar to the phrase class n-gram model, top level n-gram probabilities are used to provide probability constraints between phrases and filler words not covered by phrase grammars. The probabilities used for phrases are based on the generic rule-start probabilities and the rule-internal probabilities described in section 4.4.3. In this thesis, we will experiment with context-dependent rule-start probabilities across phrase rule boundaries while applying the shallow parsing strategy.

5.3.2 Deriving Phrase-level Shallow Parsing Grammar

Our supra-lexical linguistic modeling study is based on the TINA natural language understanding system, and the shallow parsing grammar we use is derived directly from the full TINA grammar. This ensures that the phrase level structural constraints used in speech recognition are consistent with natural language understanding. The derived shallow parsing grammar has a two-layer structure. The top layer allows connections between arbitrary phrases and filler words not covered by phrase level grammars. The bottom phrase layer represents possible word realizations of phrases. The hierarchical structure within each phrase is not preserved for shallow parsing, because most phrases we model correspond to the semantic-oriented categories in the original TINA parse tree, which are usually located at lower levels without deep hierarchy within the phrase. Furthermore, using a two-layer representation simplifies the shallow parsing structure, which facilitates the application of context-dependent probabilities using a layered FST approach.

In Wang's work [108], the phrase level grammars are also derived from the natural language grammar, except that the detailed hierarchical structures within each phrase are kept, and the generic rule-start probabilities are used with the hierarchical phrase structure. This model can be conveniently represented by RTNs; however, it would be difficult to explore context-dependent probabilities with such a framework.

The detailed approach to deriving the shallow parsing grammar can be summarized as follows. First, a set of meaning-carrying categories are manually chosen from the full-fledged TINA grammar as phrase categories. As was discussed before, the idea is to impose

early structural constraints in speech recognition for key phrases, which are essential in the subsequent meaning generation by natural language understanding. Some of the commonly used phrases such as "thank_you" and "goodbye" are also included. Next, a large training corpus is parsed according to the original TINA grammar, and the phrases covered by the chosen set of phrase categories are identified. Then, the shallow parsing grammar is constructed based on the phrases that occurred in training. The intermediate hierarchy below phrases is not preserved, resulting in a two-level shallow parsing grammar. At the top level, the phrases and individual words are allowed to have arbitrary connections. At the bottom level, word realizations of phrases are specified by the generated phrase rules. Figure 5.3.2 shows an example parse tree according to such a shallow parsing grammar.

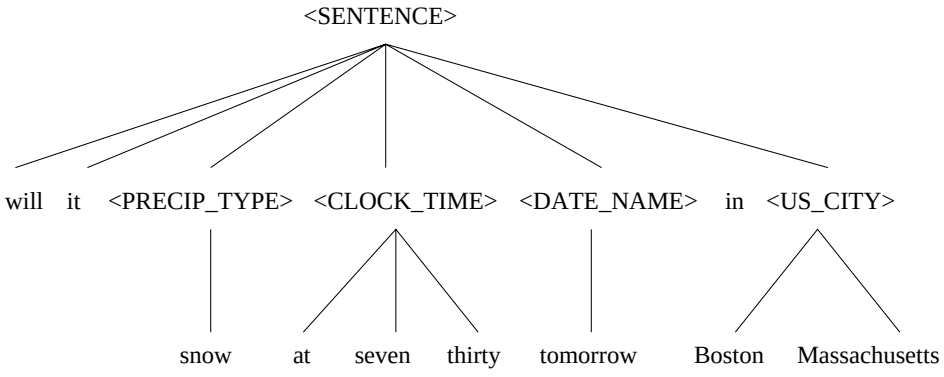


Figure 5-2: Example two-layer parse tree according to the two-level shallow parsing grammar.

It is also possible to create the shallow parsing grammar directly from the original TINA grammar by substituting the intermediate grammar rules within the phrase hierarchy. However, it can be costly to do so because of the complexity of the original grammar. The advantage of the corpus-based shallow parsing grammar derivation approach is that it eliminates rarely occurring phrases, resulting in a more compact phrase structure. However, it also has the risk of not being able to identify some legitimate phrases according to the original grammar. Such a concern is not a serious problem in our case, because most phrase structure is simple, with good support from the training corpus. Legitimate phrases not observed in training can still be accepted by the shallow parsing grammar as filler

words, which are supported by the top level n-gram probability model. The details of the probability framework used with the shallow parsing grammar are presented below.

5.3.3 Probability Framework with Shallow Parsing Grammar

The probability framework associated with the shallow parsing grammar has two major components: the top level n-gram probabilities, and the phrase level rule-start and rule-internal probabilities.

To build the top-level n-gram model, the training sentences are first reduced to phrase tags and filler words according to the shallow parsing grammar. For example, the sentence "will it snow at seven thirty tomorrow in Boston Massachusetts" shown in Figure 5.3.2 can be reduced to "will it <PRECIP_TYPE> <CLOCK_TIME> <DATE_NAME> in <US_CITY>," after it is parsed. Then, the top-level n-gram probability model is trained from the reduced sentences. Since the shallow parsing grammar allows arbitrary phrase and filler word connections at the top level, it is important to have this n-gram model to impose probability constraints over the reduced sentences.

Similar to the context-dependent probability model used with the original TINA grammar, which was discussed in section 4.4.3, there are two types of probabilities at the phrase level: rule-start probabilities and rule-internal probabilities.

The rule-start probability is specified for a phrase-start node in the two-layer shallow parse tree. The probability of the phrase-start node is conditioned not only on its parent phrase, but also on the left phrase or filler word. Such a context-dependent rule-start probability is able to capture context-dependency beyond the current phrase boundary. For example, the probability of the word "tomorrow" in Figure 5.3.2 is defined by the conditional probability $P($ tomorrow | <DATE_NAME>, <CLOCK_TIME> $)$. The previous phrase label "<CLOCK_TIME>" can help predict the word "tomorrow", better than the previous words "seven" and "thirty", which would be used in a regular word tri-gram model. We also experimented with the generic rule-start probability, which is the probability of the phrase-start node conditioned only on the parent phrase. Such generic rule-start probabilities have less detailed conditional context, and do not model context-dependency across phrase rule boundaries.

The rule-internal probability is specified for a phrase-internal node. It is defined as the probability of a phrase-internal node conditioned on the current phrase and the previous phrase-internal node. For example, the probability of the word "Massachusetts" in Figure 5.3.2 is defined by the conditional probability $P($ Massachusetts $|$ <US_CITY>, Boston $)$. The probability of a complete phrase is defined by multiplying the rule-start probability for the first word in the phrase, and the rule-internal probabilities for the rest of the words.

The rule-start and rule-internal probabilities are trained by parsing a training corpus according to the shallow grammar. With the context-dependent rule-start probabilities, adjacent phrase or word context is taken into account when estimating phrase probabilities. The complete two-layer parse tree probability can be approximated as the production of the top-level n-gram probabilities and the phrase-level rule-start and rule-internal probabilities.

5.3.4 Summary

In this section, we have proposed a shallow parsing approach to impose phrase structure constraints in speech recognition. The shallow parsing grammar is derived directly from the full natural language grammar, which can be used to impose supra-lexical recognition constraints consistent with natural language understanding. A probability framework is also designed to augment the shallow parse grammar. Such a framework combines top-level n-gram probabilities with phrase-level probabilities, and provides probability support for balancing grammar coverage with tight structural constraints. The context-dependent rule-start probabilities can model context-dependency beyond the current phrase boundary, which has the potential to offer stronger predictability.

The shallow parsing approach produces a regulated two-layer parse tree, which facilitates the application of context-dependent rule-start probability models in a layered FST framework, similar to what has been used for context-dependent hierarchical sub-lexical modeling discussed in chapter 3. In the next section, we study the integration of the shallow-parsing-based linguistic model into speech recognition using such a layered FST framework.

5.4 Context-dependent Probabilistic Shallow Parsing Using Layered FSTs

As we can see, the shallow-parsing-based supra-lexical linguistic model proposed in the previous section shares some fundamental similarities with the hierarchical sub-lexical models presented in chapter 2. They all use CFGs to impose structural constraints over basic linguistic units, and context-dependent probability models are used to augment the CFGs, providing additional probability constraints. One major difference is that, in the sub-lexical case, a complete sub-lexical structure is constructed from the grammars, which represents the full sub-lexical linguistic hierarchy. In the supra-lexical case, phrase constituents are described by phrase shallow parsing grammars, and arbitrary phrase and filler word connections are allowed at the top level. n-gram models are used to provide top level probability support. In chapter 3, we have proposed a layered approach to fold the hierarchical sub-lexical model into an FST framework. It is possible to use the same approach to build a shallow-parsing-based supra-lexical model using FSTs.

The shallow-parsing-based supra-lexical linguistic model can be integrated into speech recognition seamlessly within the FST framework. Since the shallow parsing grammars are derived from the full natural language grammar used in natural language understanding, such a tight integration has the potential of providing early feedback consistent with natural language understanding to guide the recognition search. Furthermore, as in the sub-lexical case, the unified FST framework allows global optimization to be performed on the single composed recognition search space.

As was given in expression 5.1, the FST-based SUMMIT speech recognizer we use is defined by the FST composition:

$$S \circ A \circ C \circ P \circ L \circ G \tag{5.2}$$

Our approach is to substitute the original class n-gram based supra-lexical linguistic model, which is represented by FST G, with the shallow parsing based supra-lexical model. G needs to impose phrase structure constraints specified by the shallow parsing grammar, and integrate both the top level n-gram probability models and the phrase-level rule-start and

rule-internal probability models. Similar to the sub-lexical modeling case, RTNs can be used to represent the shallow parsing grammars. The top-level n-gram model probabilities, the *generic* rule-start probabilities and the rule-internal probabilities can all be directly encoded by the transition weights within the top-level network and phrase-level networks in the RTNs. However, it is difficult to incorporate context-dependent probabilities into RTNs directly. Similar to the sub-lexical modeling approach with layered FSTs, we will use a separate FST to model the phrase-level context-dependent rule-start probabilities.

As was pointed out in section 5.2.1, there is one important concern in integrating supra-lexical linguistic models using FSTs. The standard composition and optimization procedures become computationally more expensive as the complexity of the FSTs increases. At the supra-lexical level, the vocabulary size is typically much larger than the sub-lexical phone set size, and the grammars are usually much more complex as well. In the shallow parsing approach, the full natural language grammar has been simplified to shallow parsing grammars. However, the use of FST-based context-dependent hierarchical models at the supra-lexical level can still be highly restricted due to the FST complexity.

5.4.1 FST-based Supra-lexical Model Definition

In this work, we decompose the supra-lexical model FST into two separate FSTs. The FST G in expression 5.2 is defined by the following FST composition:

$$G = R \circ T \tag{5.3}$$

where R is a weighted shallow parsing RTN, which takes in word sequences and outputs a tagged string representing the shallow parse tree. It also incorporates the probabilities that can be directly encoded within the subnetworks of R. T is the phrase-level rule-start probability FST, which encodes the context-dependent rule-start probabilities. Such a layered approach allows us to incorporate the full context-dependent probability model within the FST framework. Furthermore, it is convenient to develop the phrase-level context-dependent probability models independently, and integrate them with other probabilities using this framework. For example, we can try to use different phrase-level contexts to suit different shallow parsing grammars, which may produce a more effective context-dependent

model. The construction details of R and T are given below.

5.4.2 Shallow Parsing FST

The shallow parsing FST R is an RTN constructed from the shallow parsing grammar. The phrase grammar rules are represented by the sub-networks. R is configured to output a tagged parse string, which represents the shallow parse tree. Each phrase is enclosed by a phrase open tag and a phrase close tag. For example, the tagged parse string for the shallow parse tree given in Figure 5.3.2 is "<SENTENCE> will it <PRECIP_TYPE> snow </PRECIP_TYPE> <CLOCK_TIME> at seven thirty </CLOCK_TIME> <DATE_NAME> tomorrow </DATE_NAME> in <US_CITY> Boston Massachusetts </US_CITY>." Such a tagged string is used by the phrase-level rule-start probability FST T to apply context-dependent rule-start probabilities.

Unlike the sub-lexical parsing RTN, which is unweighted, the supra-lexical parsing RTN incorporates top-level n-gram probabilities, as well as phrase-level probabilities, including the *generic* rule-start probability and the rule-internal probability. This is because they all can be directly encoded by the transition weights in the top-level n-gram network and the phrase sub-networks. The overall structure of the parsing RTN R is similar to the FST-based class n-gram model illustrated in Figure 5-1, except that the weighted sub-networks representing the trivial word class rules are substituted by the sub-networks representing the shallow phrase rules. R can be used by itself as a supra-lexical linguistic model, with top-level n-gram probabilities, and phrase probabilities without applying context information beyond the current phrase.

5.4.3 Phrase-level Rule-start Probability FST

The phrase-level rule-start probability FST T is constructed to apply context-dependent rule-start probabilities. The probability of a rule-start node is conditioned on the current parent and its left sibling. The basic context transition of T is designed as follows. It starts with the state representing the current conditional context, i.e., the current parent "P" and its left sibling "a". Then, it applies the context-dependent rule-start probability as it makes the transition accepting the rule-start node "m". Next, it filters out the phrase internal

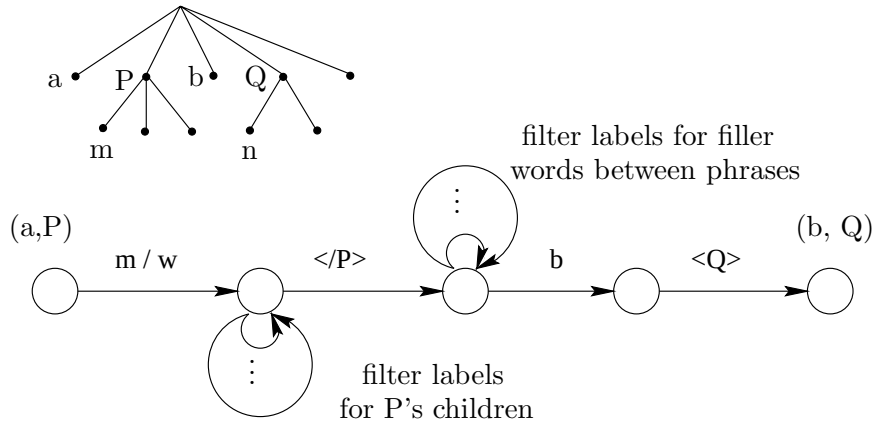


Figure 5-3: The context state transition diagram in the phrase-level context-dependent rule-start probability FST. It shows the transitions from a state (a,P) to state (b,Q), where P, a, Q, and b are the current parent, the current parent's left sibling, the next parent, and the next parent's left sibling in the shallow parse tree, respectively. "w" is the context-dependent rule-start probability Prob(m | a, P) normalized by the generic rule-start probability Prob(m | #, P), where "m" is the first child of P, and "#" is the rule-start marker.

nodes and the filler words between phrases, and finally reaches the state representing the next conditional context, i.e., the next parent "Q" and its left sibling "b". Figure 5-3 illustrates such a context transition.

Given the basic context transition described above, the context states of T are connected by such context transitions for each trained rule-start probability instance. The overall structure of FST T is similar to a regular n-gram FST, as illustrated in Figure 3-4. Since T is composed with R, which already applied the generic rule-start probability, the context-dependent rule-start probability applied in T is normalized by removing the corresponding generic rule-start probability. As was discussed in section 4.5.2, ill-trained rule-start probabilities without sufficient observations need to be pruned. In this case, we back off to the generic rule-start probability without using context information beyond the current rule. This back-off strategy yields a more robust phrase probability estimation.

5.4.4 Construction of the Complete Model

Our approach to constructing the complete shallow-parsing-based supra-lexical linguistic model using FSTs can be summarized as follows. First, a set of key semantic categories

are specified in the full TINA natural language grammar. In this work, these categories are manually selected to reflect key semantic phrases. It is also possible to select the categories automatically according to some optimization criteria, such as the final model complexity. Then, a large training corpus is parsed using the original grammar. The phrases are identified, and the training sentences are reduced to a sequence of phrase tags and filler words between the phrases. The reduced sentences are used to train a top-level bi-gram model, while the identified phrases are used to generate the shallow parsing grammar. Next, the rule-internal probabilities, the generic rule-start probabilities, and the context-dependent rule-start probabilities are trained according to the shallow parsing grammar. Ill-trained context-dependent probability models are pruned. After training the probability models, the top-level n-gram probabilities, the generic rule-start probabilities, and the rule-internal probabilities are used to construct the weighted shallow parsing RTN R according to the shallow parsing grammar. The context-dependent rule-start probabilities are used to construct the phrase-level context-dependent rule-start probability FST T. Finally, R and T are composed to obtain the complete supra-lexical linguistic model. Figure 5-4 illustrates such an approach.

In principle, R and T can be pre-composed and optimized. However, such operations become much more computationally expensive as the complexity of R and T increases. The most problematic operation is the determinization step during optimization. Since there are a lot of weighted ϵ transitions used for smoothing in the top-level n-gram model, it is difficult to determinize the composed FST $R \circ T$. Furthermore, the supra-lexical model FST $R \circ T$ needs to be composed with the rest of the FSTs representing other knowledge sources, and the complete search space needs to be optimized to achieve efficient recognition search. Such optimization also becomes difficult. In fact, if we were to use a tri-gram model as the top-level probability model, the complete speech recognition search space could not be pre-composed and optimized in advance, given the computational resources we currently use. This is a major limitation in applying such hierarchical supra-lexical linguistic models with context-dependent probabilities.

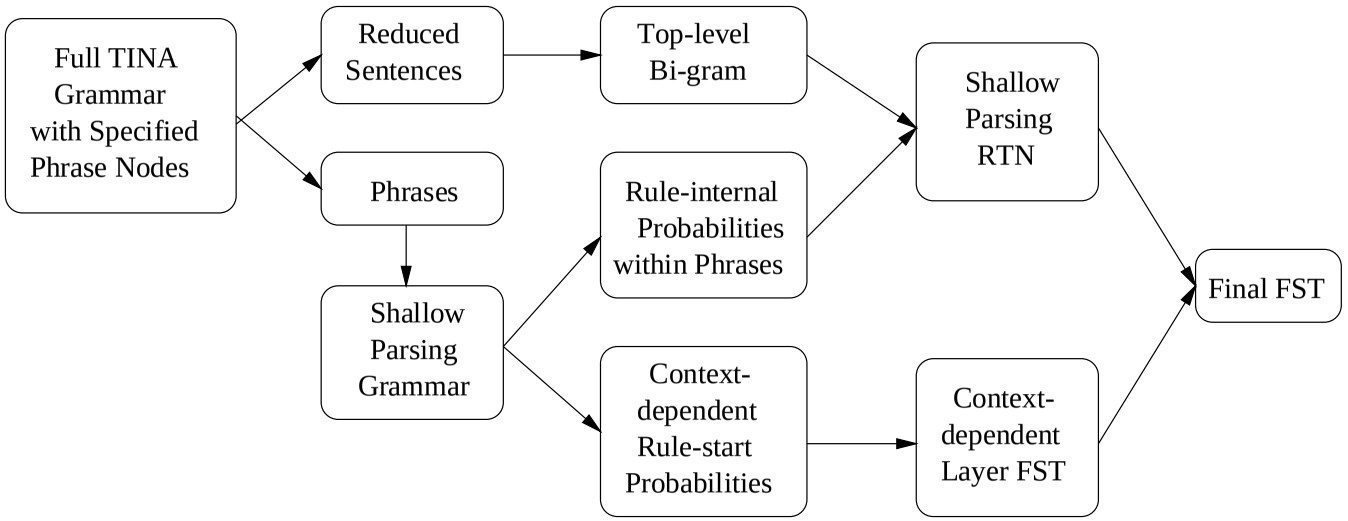


Figure 5-4: Construction of FST-based linguistic models derived from TINA language understanding system.

5.4.5 Recognition Experiments

We have experimented with the shallow-parsing-based supra-lexical linguistic model in the JUPITER weather information domain. Table 5-1 summarizes the recognition word error rate (WER) results for four different systems: (1) standard class bi-gram, (2) shallow parsing with *generic* rule-start probabilities (using FST R), (3) shallow parsing with *context-dependent* rule-start probabilities (using FST $R \circ T$), and (4) standard class tri-gram. Bi-gram models are used as the top-level probability model in systems (2) and (3). The four recognizers are tested on the full test set and the in-vocabulary subset, which have been used for sub-lexical linguistic modeling experiments. The context-dependent rule-start probability models trained from insufficient training observations are pruned, and the pruning threshold is determined through an independent development set.

Supra-lexical Linguistic Model	*WER on Full Test Set (%)*	*WER on In-vocabulary Test Set (%)*
Class Bi-gram	17.0	12.6
Generic Rule-start Probabilities (FST R only)	16.8	12.1
Context-dependent Rule-start Probabilities (FST $R \circ T$)	16.3	11.8
Class Tri-gram	15.3	11.0

Table 5-1: The recognition word error rate (WER) results in the JUPITER domain on the full test set and the in-vocabulary subset.

We can see from the results that, compared to the baseline class bi-gram model, the proposed shallow parsing model with top-level bi-gram and generic-rule start probabilities is able to reduce word error rates on both test sets. The use of context-dependent rule-start probabilities further improves recognition. On the full test set, the WER is reduced from 17.0% to 16.3%, with a relative WER reduction of 4.1%. On the in-vocabulary test set, the WER is reduced from 12.6% to 11.8%, with a relative WER reduction of 6.3%.

This suggests that the context-dependent shallow parsing approach with top-level n-gram probabilities can offer phrase structure constraints supported by context-dependent phrase probabilities, and may achieve a lower WER compare to a class n-gram model with the same order. We also see a better recognition improvement on the in-vocabulary test set than on the full test set. One possible reason is that the shallow parsing rules do not contain unseen words; therefore, phrase structures can be better modeled by the shallow parsing rules on the in-vocabulary test set.

We also performed the McNemar significance test [105] to obtain the significance levels for system (2) and system (3) against the baseline system (1). The significance level (denoted P) evaluates the hypothesis that two recognizers are the same in terms of correctly recognized words. A low value of P implies that the WER difference between two recognizers is statistically significant. Table 5-2 shows the McNemar test results.

Supra-lexical Linguistic Model	*Significance Level on Full Test Set*	*Significance Level on In-vocabulary Test Set*
Generic Rule-start Probabilities (FST R only)	0.168	0.120
Context-dependent Rule-start Probabilities (FST $R \circ T$)	0.0063	0.0006

Table 5-2: The McNemar significance levels against the baseline class bi-gram recognizer on the full test set and the in-vocabulary subset.

The very low McNemar significance levels for system (3) (using FST $R \circ T$) on the full test set (0.0063) and the in-vocabulary subset (0.0006) suggest that the WER improvements obtained using the context-dependent shallow parsing approach are statistically significant. Compared to the significance levels for system (2) (using FST R only) on the full test set (0.168) and the in-vocabulary subset (0.120), the significance levels for system (3) are much lower. This indicates the effectiveness of applying context-dependent rule-start probabilities to model local context-dependencies beyond the rule boundaries.

The proposed context-dependent shallow parsing model with top-level n-gram probabil-

ities achieves lower recognition word error rates in this experiment, compared to a regular class n-gram model with the same order. However, we have found that the FST encoding context-dependent rule-start probabilities (FST $R \circ T$) has 1100K arcs and 36K states, which is significantly larger than the FST encoding only generic rule-start probabilities (FST R) with 222K arcs and 2K states. The unstructured class bi-gram FST consists of only 58K arcs and 1.2K states. If we were to use top-level tri-gram probabilities in the context-dependent shallow parsing model, the recognition FSTs could not be pre-composed and optimized given the computation resources we currently use, though similar recognition improvements are potentially possible. Therefore, the application of the context-dependent shallow parsing model can be limited by the complexity of the FSTs.

5.5 Summary

In this chapter, we have proposed a shallow parsing approach for supra-lexical linguistic modeling. Shallow parsing can help balance sufficient generality (i.e., coverage) and tight phrase structure constraints (i.e., precision), which is particularly important in conversational interfaces where spontaneous speech has to be handled. The shallow parsing grammar is directly derived from the full natural language understanding grammar, and is augmented with top-level n-gram probabilities and context-dependent phrase probabilities. Such an approach can help impose recognition constraints consistent with natural language understanding.

The proposed context-dependent hierarchical shallow parsing model is constructed within a unified layered FST framework, which has also been used for sub-lexical hierarchical linguistic modeling. A shallow parsing RTN is built from the shallow parsing grammar, which outputs a tagged parse string representing the shallow parse tree. It also encodes top-level n-gram probabilities, phrase-level generic rule-start probabilities, and phrase-level rule-internal probabilities. A separate FST is used to model phrase-level context-dependent rule-start probabilities, and is composed with the parsing RTN to build the complete supra-lexical linguistic model. Such a unified FST framework is able to seamlessly integrate the shallow-parsing-based supra-lexical linguistic model into speech recognition. It also has the

potential of providing early feedback from natural language understanding in the speech recognition search.

Our speech recognition experiments show that the proposed context-dependent shallow parsing model with top-level n-gram probabilities and phrase-level context-dependent probabilities achieve lower recognition word error rates, compared to a regular class n-gram model with the same order. However, the final composed FST representing the speech recognition search space can be significantly larger than the regular class n-gram model. With a higher order top-level n-gram, pre-composition and optimization are highly restricted by the computational resources available, which can limit the application of such a sophisticated context-dependent supra-lexical linguistic model. However, given the potential of such models, it may be worth pursing a strategy similar to the FST-based incremental n-gram model approach [30], where the complete supra-lexical model is factored into two FSTs. The first one can be statically composed and optimized, and the second one is composed on-the-fly. This approach is able to include part of the supra-lexical linguistic model FST into early pre-composition and optimization, while keeping the complete supra-lexical model through incremental dynamic composition.

Based on our experiences of integrating supra-lexical linguistic knowledge, we believe that, without giving special consideration to reducing FST complexity, it may not be practical to construct a unified system integrating context-dependent hierarchical linguistic constraints at both sub-lexical and supra-lexical levels using the FST framework. The pre-composition operation tends to replicate the substructure of the individual FSTs, which is a major source of the increase in search space size. Although FST optimization operations, such as minimization and determinization discussed in section 5.2.1, can help collapse the replicated substructures, we find that they are not sufficient to solve the FST complexity problem. One possible solution is to dynamically compose the FSTs during recognition. However, such an approach suffers from the increase in run-time cost. Furthermore, the search space is not pre-optimized. In Section 7.2, we will present our view of systematically pruning and composing the FSTs, such that an appropriate compromise between the tightness of knowledge integration and the complexity of search space can be achieved.

Chapter 6

Dynamic Reliability Modeling in Speech Recognition

6.1 Introduction

Speech recognition can be formulated as a problem of searching for the best sequence of symbols, subject to the constraints imposed by acoustic and linguistic knowledge. In the previous chapters, we have discussed our efforts in integrating hierarchical *linguistic* knowledge at both sub-lexical and supra-lexical levels within a layered FST framework. Another important aspect of achieving more accurate and robust speech recognition is the integration of *acoustic* knowledge, which is formulated by acoustic models. Acoustic models characterize acoustic features extracted from the input speech signal. More specifically, they specify the likelihood that a particular linguistic event (e.g., a phone sequence or a word sequence) has generated the input sequence of acoustic features.

In typical speech recognition systems, acoustic constraints are applied uniformly across the entire utterance. This does not take into account the fact that some acoustic units along the search path may be modeled and recognized more reliably than others, due to differences in their acoustic-phonetic characteristics, the particular feature extraction and modeling approaches the recognizer chooses, the amount and quality of available training data, etc. One possible way to incorporate reliability information is through word-level and utterance-level rejection [82]. However, this approach provides confidence information after

the recognition phase, and as such the confidence score is usually measured from a set of chosen features, most of which are obtained after the recognition is completed [48]. To address such an issue, we have attempted to incorporate acoustic model reliability information during the recognition search phase in order to help the recognizer find the correct path.

In this chapter, we introduce the notion of dynamic reliability scoring that adjusts path scores according to trained reliability models. Such path score adjustments are made while the recognizer searches through the predefined search space, which can be viewed as the composition of an acoustic-phonetic network and a lexical network. The recognizer estimates the reliability of choosing a specific hypothesized arc to extend the current path, and adds a weighted reliability score to the current path score. With such dynamic reliability information, two immediate benefits can be obtained. First, the adjusted overall path score now reflects a more comprehensive evaluation of the complete path; thus a final path with a higher score is more likely to be a correct path. Second, with necessary pruning to balance accuracy and complexity, unpromising partial paths are pruned according to their current scores. Dynamic reliability modeling can help obtain better evaluations of partial paths as well. This is important because pruning errors are not recoverable once they happen.

In the next sections, we first introduce the architecture of the speech recognizer we use, and explain the acoustic-phonetic and lexical networks that define the recognition search space. Next, we elaborate on the details of constructing, training and applying the dynamic reliability models. Some related issues such as back-off models and iterative training are also presented. The experiments in reliability scoring are conducted on two weather information domains, JUPITER [116], which is in English, and PANDA, which is a predecessor of MUXING [108] in Mandarin Chinese. The experimental results, conclusions and future directions are also presented.

6.2 Acoustic-Phonetic and Lexical Networks

The recognizer we use for our dynamic reliability modeling study is the MIT SUMMIT [36] segment-based recognizer, with the basic pronunciation network sub-lexical model and the n-gram language model, which were discussed in section 2.2.2 and section 4.2.1, respectively.

To facilitate the introduction of acoustic reliability models, we view the recognition search space as a composition of an acoustic-phonetic network and a lexical network, with n-gram language model scores applied during the recognition search. Since we focus on acoustic model reliabilities, language model scores are separated from acoustic model scores while training the reliability models.

As was given in expression 2.5, the recognition search space is defined by the FST composition:

$$S \circ A \circ C \circ P \circ L \circ G \tag{6.1}$$

where S represents an *acoustic-phonetic network* built from the input speech signal. It provides possible segmentations of the speech waveform, according to observed signal landmarks. Corresponding segment or boundary features are extracted based on the hypothesized segmentations, and these features are used to obtain acoustic scores from acoustic models represented by A. Since A can be established for context-dependent phones, e.g., di-phones and tri-phones, C is used to map context-dependent phone labels to regular phone labels. $P \circ L$ is viewed as a *lexical network* mapping phones to words, and is established from the recognizer's vocabulary. Each word in the vocabulary is represented by a pronunciation network, which is constructed by applying phonological rules to the baseform network of each word. These networks are then combined into a single lexical network by connecting word-end nodes and word-start nodes satisfying the inter-word pronunciation constraints. Such a lexical network encodes possible phone sequences the recognizer can choose from. G imposes additional n-gram probability constraints on word sequences.

Given the composed acoustic-phonetic and lexical network, the recognizer essentially performs a search in such a predefined search space, finds the best path, and gives the corresponding word sequence as the result. Our main objective is to integrate acoustic model reliability knowledge during the recognition search, dynamically adjusting partial path scores to reflect the additional acoustic model reliability information.

6.3 Acoustic Reliability Models

In this section, we describe the acoustic reliability models and the training procedures we use. The notion of reliability here refers to how confident we are while choosing a specific hypothesized arc to extend the current partial path during the recognition search. It could happen that a high scoring arc is actually not in the correct path, particularly if its immediate competitors have similar high scores. In this case it is less confident to choose this arc, even though it has a relatively high acoustic score. On the other hand, an arc with relatively low acoustic score could be the right candidate to extend the current path if its competitors have much lower scores.

We construct reliability models to formulate such notions of acoustic model reliability. The reliability models offer a reliability estimation for extending the current partial path using one specific arc as opposed to using its immediate competing alternatives in the composed acoustic-phonetic and lexical network. These reliability measurements are then used to adjust the partial path scores and help the recognizer better evaluate the partial paths.

6.3.1 Reliability Model Definition

The reliability models we propose in this work are associated with the phone arcs in the lexical network. For each lexical phone arc, separate Gaussian models are built to describe the acoustic scores obtained when it corresponds to an arc in the *correct* path from the composed acoustic-phonetic and lexical network, and the acoustic scores obtained when it corresponds to a *competing* arc not in the correct path. Such trained Gaussian models are used as the reliability models, which characterize typical acoustic scores for a lexical arc in the different situations mentioned above. If a lexical arc is reliably modeled, the two Gaussian models would have a relatively large difference in their means, and relatively small variances. We choose to use Gaussian models here mainly because of their simplicity. More complex models, such as mixture Gaussian models, can be used to better characterize the acoustic score distributions. However, there will be more model parameters to be trained, and more training data are required to obtain a robust model.

An important aspect of our approach is that all the acoustic scores used to train the reliability models are normalized by a "catch-all" model. As we know, the *a posteriori* probability score of a class ω_i given the feature observation x is:

$$p(\omega_i|x) = \frac{p(x|\omega_i)}{p(x)}p(\omega_i) \tag{6.2}$$

In most speech recognition systems, only the acoustic likelihood score $p(x|\omega_i)$ is used for comparing different hypotheses given the observation x. Since $p(x)$ remains the same for a given x, such a simplification does not affect the comparison. However, the acoustic likelihood scores are not directly comparable across different feature observations. Therefore, it is not suitable to train the reliability models directly from such unnormalized scores.

In this work, we use a normalized log-likelihood scoring scheme, and the acoustic score is given by:

$$\log(p(x|\omega_i)/p(x)) \tag{6.3}$$

where x is the feature observation, $p(x|\omega_i)$ is the probability density of x given the class model ω_i, and $p(x)$ is the catch-all normalization model defined as:

$$p(x) = \sum_{j=1}^{N} p(x|\omega_j)P(\omega_j) \tag{6.4}$$

The normalized log-likelihood is expressed in the log domain and can be viewed as a zero-centered score. An acoustic score greater than zero represents a greater than average possibility that the feature observation belongs to the hypothesized class. The use of normalized acoustic scores ensures that the reliability models are built from comparable acoustic scores. This is essential for characterizing the score differences for correct arcs and competing arcs.

6.3.2 Training Approach for Reliability Models

The training approach for the reliability models can be summarized as follows. First, given training utterances transcribed at the word level, a standard forced alignment is

conducted, which searches the best path in a search space specified by the known word sequence. Such a search space is much more constraining compared to the original composed acoustic-phonetic and lexical network, since only the word sequence given by the word-level transcription is allowed for each utterance. The forced alignment results are used as references to correct paths. Then, for each partial path along the forced path in the original composed network, the acoustic score of the arc extending the forced path, denoted s, and the acoustic scores of the immediate competing arcs that are not in the forced path, denoted $t_1, t_2, \ldots, t_n$, are collected. Note that the acoustic scores are the normalized log-likelihood scores, as discussed in section 6.3.1. After that, for each arc in the lexical network, Gaussian models for correct scoring (i.e., scores of corresponding arcs that are in the forced path from the composed acoustic-phonetic and lexical network), M_s, and incorrect scoring (i.e., scores of corresponding arcs not in the forced path), M_t, are trained from the collected acoustic scores.

Figure 6-1 shows an example of trained reliability models M_s and M_t, which are associated with the arc labeled [t] for the word "want" in the lexical network. In general, M_s is centered at a score greater than zero with smaller variance, while M_t is centered at a score less than zero with greater variance. The further apart the two models and the smaller their variances are, the easier it would be to distinguish the correct arc from its immediate competing alternatives while searching the composed acoustic-phonetic and lexical network.

6.4 Dynamic Reliability Scoring

In this section, we describe the application of the reliability models in speech recognition, and some related issues such as back-off models and iterative training. After the reliability models for each arc in the lexical network are trained, we can use these models to obtain a reliability score while extending the current partial path using a specific hypothesized arc. This reliability measurement helps the recognizer adjust the current path score according to the additional reliability information, and better evaluate the partial paths.

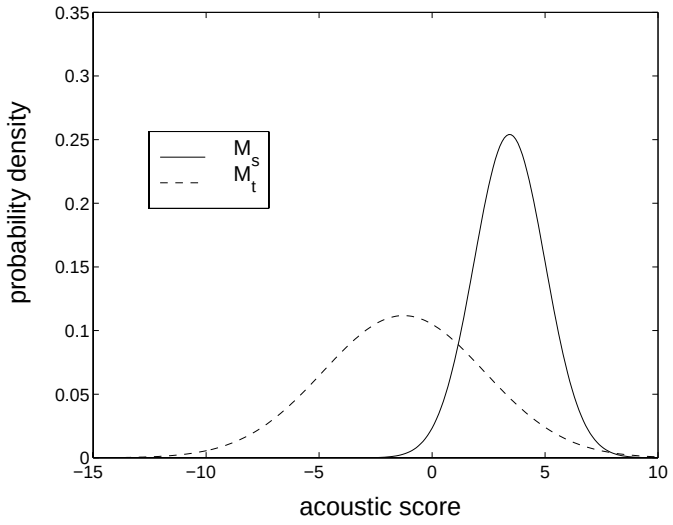


Figure 6-1: The trained reliability models M_s (correct scoring) and M_t (incorrect scoring) associated with the arc labeled [t] for the word "want" in the lexical network.

6.4.1 Reliability Measurement Definition and Application of Reliability Models

The reliability measurement we use is essentially the likelihood that a particular candidate arc in the composed network with acoustic score s is actually in the right path, while its immediate competitors with acoustic scores $t_1, t_2, \ldots, t_n$ are not in the right path. Assuming the correct path and the competing alternative paths are independent of each other, such a likelihood can be represented by the probability density score p defined in the following equation:

$$\begin{aligned} p &= p(s|M_s)p(t_1, t_2, \ldots, t_n|M_t) \\ &= p(s|M_s)\prod_{i=1}^{n} p(t_i|M_t) \end{aligned} \tag{6.5}$$

where M_s represents the correct scoring reliability model for the candidate arc extending the current path, and M_t represents the incorrect scoring reliability models corresponding to the competitors. It can be further rewritten as the following equation using $p(s|M_t)$ as a normalization factor:

$$
\begin{aligned}
p &= \frac{p(s|M_s)p(s|M_t)\prod_{i=1}^{n} p(t_i|M_t)}{p(s|M_t)} \\
&= \frac{p(s|M_s)}{p(s|M_t)} p(s, t_1, t_2, \ldots, t_n|M_t) \qquad (6.6)
\end{aligned}
$$

where $p(s|M_s)$ and $p(s|M_t)$ are the probability densities of having acoustic score s for the candidate arc given the *correct scoring* and *incorrect scoring* reliability models, respectively. Since we use reliability scores to help the recognizer choose a candidate arc from its *immediate* competing arcs, $p(s, t_1, t_2, \ldots, t_n|M_t)$ in equation 6.6 is a constant given all the directly connected arcs extending the current path. Therefore, we can just use the log domain score $\log(p(s|M_s)/p(s|M_t))$ as the reliability measurement. It represents a normalized, zero-centered reliability score. For an arc with a relatively high acoustic score, if it usually has similar high scores as competing arcs, the reliability scoring can give a negative result. On the other hand, for an arc with a relatively low acoustic score, if it usually has much lower scores as competing arcs, the reliability scoring can give a positive result. Such a simplification also saves a lot of computational effort during the recognition search.

The log domain reliability score can be linearly combined with the acoustic model score to adjust the current partial path score. The optimal combination weights are determined on a development set. The adjusted partial path scores can help reduce pruning errors during search. Also, with the integrated acoustic model reliability information, a complete path with a better overall path score is more likely to be the correct path.

6.4.2 Back-off Models

The reliability models M_s and M_t are typically trained for each arc in the lexical network. However, some arcs in the lexical network do not have enough training observations; therefore, the corresponding models can be ill-trained. To alleviate such a sparse data problem, we have established two types of back-off models, namely the *phonetic* back-off model and the *generic* back-off model. The phonetic back-off model is trained by combining the training data for all the arcs bearing the same phone label. The generic back-off model is trained

from all the training data available regardless of their corresponding lexical arcs or phone labels.

The original arc-specific models, the phonetic back-off models and the generic back-off models can be linearly combined to obtain more robust reliability model scores, as defined by the following equation:

$$r_1 \frac{p(s|M_s)}{p(s|M_t)} + r_2 \frac{p(s|M_s^P)}{p(s|M_t^P)} + r_3 \frac{p(s|M_s^G)}{p(s|M_t^G)} \tag{6.7}$$

where M, M^P and M^G are the original arc-specific model, the phonetic back-off model and the generic back-off model, respectively; r_1, r_2 and r_3 are the weights for the original models and the back-off models, satisfying $r_1 + r_2 + r_3 = 1$. These weights can be adjusted and optimized on a development set. In this work, we find the following simplified approach satisfactory to determine the weights. First, the phonetic back-off model and the generic back-off model are combined, with combination weights set proportional to the amount of training data available for each type of model. Then the original arc-specific model is interpolated with the combined back-off model, with weights proportional to the amount of training data for the arc-specific model and the phonetic back-off model.

6.4.3 Iterative Training

The training utterances are transcribed at the word level. Since it is costly to transcribe the utterances manually at the phonetic level, forced alignment is a commonly used approach to obtain reference phonetic transcriptions. With the addition of dynamic reliability scoring, we can potentially obtain more accurate phonetic transcriptions and reference paths during forced alignment, which can help improve the acoustic and reliability models.

The iterative training procedure for the acoustic models and reliability models is as follows. First, given the training utterances with word-level transcriptions, a forced search is conducted using the original acoustic models. Then, the initial reliability models are trained according to the forced paths. New acoustic models can also be trained based on the resulting phonetic alignments. Next, the trained reliability models and the new acoustic models can be used to obtain new forced phonetic alignments. After that, reliability models

and acoustic models are updated according to the new alignment results. Such a procedure can improve the forced alignment results iteratively, and help build better acoustic models and reliability models.

In practice, we find that the recognition performance converges quickly after a few iterations. More details are given in the following section, presenting the experimental results.

6.5 Experimental Results

We have incorporated the reliability scoring scheme into the segment-based, SUMMIT [36] speech recognition system, which can use both segment acoustic models and di-phone acoustic models. The segment acoustic models are trained from acoustic features extracted for complete phone segments. They do not model the phonetic context of neighboring phones. The di-phone acoustic models are trained from acoustic features extracted at the possible boundaries of phone segments. They model the transition features from the previous phone to the next phone, which integrates phonetic context information. However, the number of di-phone models is generally much larger than the number of segment models. Therefore, di-phone models require more training data than segment models. They can perform better than segment models if enough training data are available.

The recognizer's acoustic models and reliability models are trained and evaluated in two application domains, selected to cover situations with both a large and a limited training corpus. The first one is the JUPITER weather information domain in English, and the second one is the PANDA weather information domain in Mandarin Chinese, which is a predecessor of MUXING [108]. For the JUPITER domain, the training set consists of 24,182 utterances, and the test set consists of 1,806 utterances. Note that the training and test sets are from a previous version of the JUPITER corpus available when this experiment was conducted. The corpus is different from the more recent version used in the sub-lexical and supra-lexical linguistic modeling experiments. For the PANDA domain, the training set consists of 1455 utterances, and the test set consists of 244 utterances. The acoustic models and reliability models are trained using the iterative training approach discussed in section 6.4.3. For

the JUPITER domain, both di-phone models and segment models are used. The reliability models are built on the combined di-phone and segment acoustic scores. For the PANDA domain, due to insufficient data for training di-phone models, only segment models are used, and therefore the reliability models are built on segment acoustic scores. Utterances with out-of-vocabulary (OOV) words are excluded from training and test sets in both domains, because the basic pronunciation network sub-lexical model we use does not handle OOV words. It is of course possible to apply a similar reliability scoring approach with more sophisticated sub-lexical models that support unseen words, for example, the hierarchical sub-lexical model discussed in chapter 2.

Tables 6-1 and 6-2 show the recognition word error rate (WER) results before and after applying reliability scoring in the JUPITER and PANDA domain. Note that the baseline JUPITER WER results are different from those presented in the previous chapters, because the JUPITER corpus used in this experiment is an earlier version, as was mentioned above. Three iterations of training are performed, and the recognition results are given for each iteration.

Iteration Number	*WER without Reliability Models(%)*	*WER with Reliability Models(%)*	*Relative WER Reduction (%)*
1	12.1	10.1	16.5
2	10.3	9.2	10.7
3	10.2	9.2	9.8

Table 6-1: The recognition WER results in the JUPITER domain with three iterations of training, with and without applying the dynamic reliability models. Relative WER reductions after applying the dynamic reliability models are also shown.

Iteration Number	*WER without Reliability Models(%)*	*WER with Reliability Models(%)*	*Relative WER Reduction (%)*
1	11.9	9.8	17.6
2	10.4	8.9	14.4
3	9.7	8.5	12.4

Table 6-2: The recognition WER results in the PANDA domain with three iterations of training, with and without applying the dynamic reliability models. Relative WER reductions after applying the dynamic reliability models are also shown.

From the results, we can see that the reliability modeling approach is able to reduce the recognition errors in both domains for all the three training iterations. In the JUPITER domain, after three iterations of training, the WER is reduced from 10.2% to 9.2% using dynamic reliability scoring, with a relative WER reduction of 9.8%. We can also see that the WERs decrease monotonically after each iteration, which suggests that the iterative training approach is effective in improving the acoustic and reliability models. It is also shown that the WERs converge quickly with the three training iterations. In the PANDA domain, the WER is reduced from 9.7% to 8.5%, with a relative WER reduction of 12.4%, also after three iterations of training. In this experiment, the relative WER reduction in the PANDA domain is higher than that in the JUPITER domain, possibly due to the limited and less variant utterances in the PANDA corpus.

Figure 6-2 gives an example of the recognition search results for a test utterance with and without the reliability models. The corresponding best paths in the acoustic-phonetic networks are highlighted. Phonetic-level and word-level alignments are also shown. In this example, the reliability models are able to help correct the hypothesized path and recognize the word sequence "Massachusetts please".

6.6 Discussions

In this chapter, we have proposed a dynamic reliability modeling approach that integrates acoustic model reliability information directly in the recognition search. Reliability models are used to formalize the notion that, along the recognition search path, some acoustic units may be modeled and recognized more reliably than others. Typical confidence scoring and rejection methods also address this problem; however, confidence scores are usually obtained after the completion of the recognition search, and therefore do not alter the final recognition results. In contrast, the proposed dynamic reliability scoring method integrates the reliability information on-the-fly during the recognition search, which can help early recovery of recognition errors. We have shown that dynamic reliability modeling is able to improve the recognition accuracy in both the JUPITER and PANDA domain. Furthermore, it is demonstrated that the iterative training approach is helpful to improve the acoustic

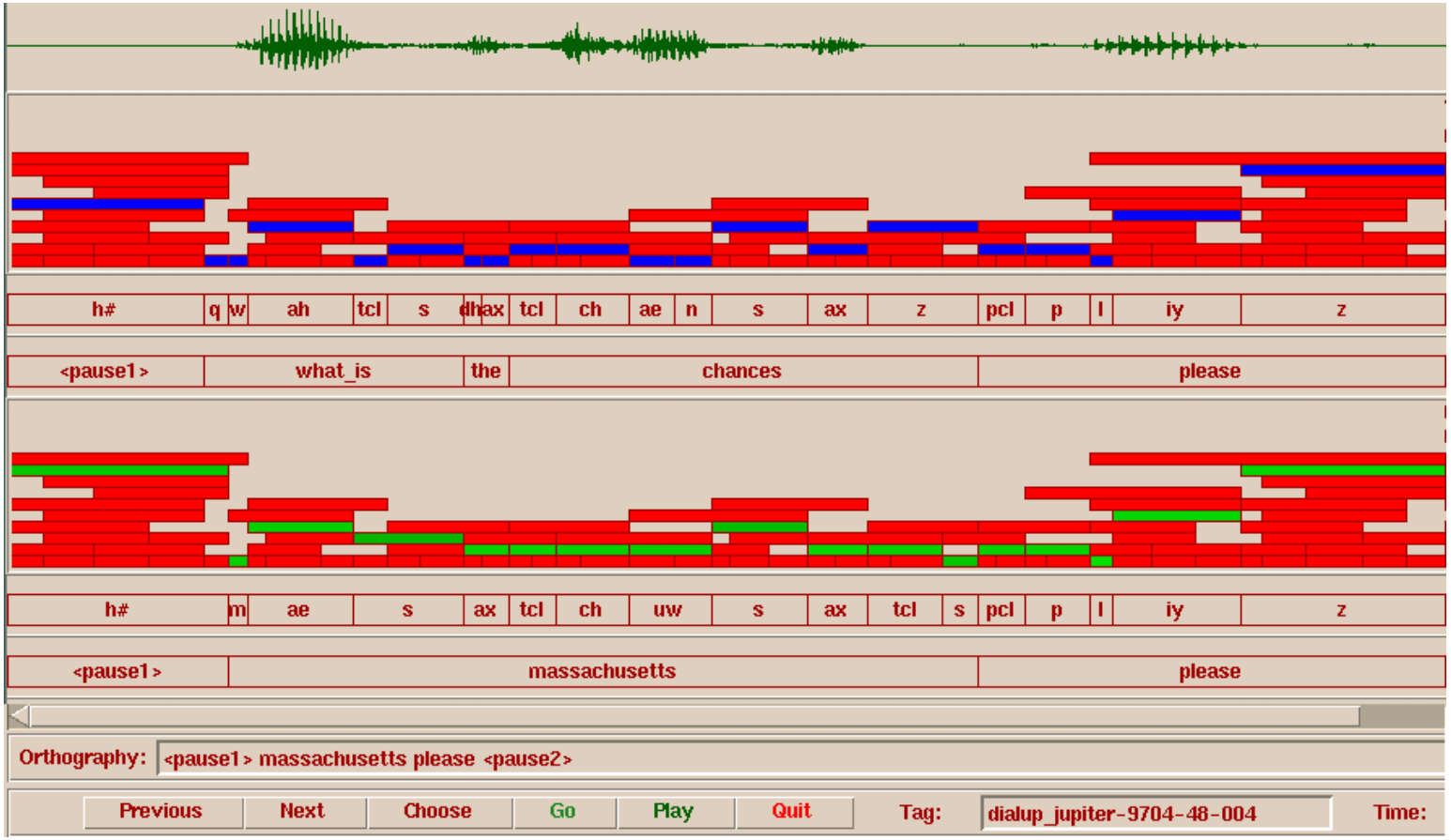


Figure 6-2: Example recognition search results with and without reliability models in the JUPITER domain. The upper panel shows that, without reliability models, the utterance is incorrectly recognized as "what is the chances please". The lower panel shows the result using the reliability models, and the utterance is correctly recognized as "Massachusetts please". The corresponding best paths in the acoustic-phonetic networks are highlighted.

models and the reliability models.

Currently the reliability models are established based on the candidate arcs directly connected to the current partial path. The reliability scores are only used to help the recognizer choose the best arc from its *immediate* competitors. Ideally, we would like to provide reliability measurements based on all the future arcs extending the current path. Since this may be highly computationally expensive, a possible compromise is to consider extending the current partial path within a certain interval. This would help reduce the bias of only considering the immediate future context, thus giving better reliability adjustments.

Note that the dynamic reliability modeling needs to compute reliability scores during recognition search, which increases computation. In applications where fast response is crucial, dynamic reliability models can be used to improve forced alignments only. More accurate alignments can result in improved acoustic models, which benefits recognition. This approach does not impose additional computation during recognition, although the full capacity of reliability modeling may not be reached.

We also notice that, with a given recognition lexicon, it is usually not necessary to accurately recognize every acoustic unit in the word to get correct recognition results. It is possible to use only several reliable pieces of the word to distinguish it from other words. Therefore, it can be beneficial to obtain such reliable pieces with the guidance of the reliability measurements, and change the lexical access scheme from precise matching to reliable-island matching. Such an island-driven lexical access approach has the advantage of using reliably modeled and less variant phones to characterize words with complex pronunciation variations. It can also help address the non-native speech [64] problem.

Chapter 7

Summary and Future Directions

7.1 Thesis Contribution and Summary

Speech-based conversational interfaces have received much attention as a promising natural communication channel between humans and computers. A typical conversational interface consists of three major systems: speech understanding, dialog management and spoken language generation. In such a conversational interface, speech recognition as a front-end of speech understanding remains to be one of the fundamental challenges for establishing robust and effective human/computer communications. On the one hand, the speech recognition component in a conversational interface lives in a rich system environment. Diverse sources of knowledge are available and can potentially be beneficial to its robustness and accuracy. For example, as was discussed in Chapter 4, the natural language understanding component can provide linguistic knowledge in syntax and semantics that helps constrain the recognition search space. On the other hand, the speech recognition component in a conversational interface also faces the challenge of spontaneous speech, and it is important to address the casualness of speech using the knowledge sources available. For example, the sub-lexical linguistic information discussed in Chapter 2 would be very useful in providing generic sub-word structural support not limited to the system's vocabulary, and the dynamic reliability information discussed in Chapter 6 may help improve recognition robustness for poorly articulated speech.

In general, we were interested in the integration of different knowledge sources in speech

recognition to improve its performance. In this thesis, we mainly focused on the integration of knowledge sources within the speech understanding system of a conversational interface. More specifically, we studied the formalization and integration of hierarchical linguistic knowledge at both the sub-lexical level and the supra-lexical level, and proposed a unified framework for integrating hierarchical linguistic knowledge in speech recognition using layered finite-state transducers (FSTs). We also studied dynamic reliability modeling that integrates knowledge of acoustic model reliability in speech recognition. The main contributions of this thesis can be summarized as follows:

- **Proposed a unified framework integrating hierarchical linguistic constraints into speech recognition using FSTs.**

  Hierarchical linguistic knowledge is available at both the sub-lexical level and the supra-lexical level, and can help improve speech recognition. However, the lack of a general framework to express and apply such knowledge is one important problem that hinders the effective use of hierarchical linguistic constraints in speech recognition. The linguistic hierarchy is typically represented by context-free grammars (CFGs), and can be augmented with context-dependent probability models beyond the standard CFG formalism. Separately implemented probabilistic parsers can be used for hierarchical linguistic modeling. However, they need to directly interact with the recognition search control using specially designed interfaces. Such an architecture is difficult to re-configure or extend to suit a wider variety of applications. In this thesis, we have proposed a unified framework using layered FSTs, which can be used to seamlessly integrate hierarchical linguistic knowledge in an FST-based speech recognizer. In this framework, CFG rules are compiled into recursive transition networks (RTNs) for parsing, and the RTNs are composed with a series of weighted FSTs to capture the context-dependent probabilities across rule boundaries at different levels of the linguistic hierarchy. The complete hierarchical linguistic model is then composed with other recognition FSTs to construct a uniform search space integrating various knowledge sources. Such a unified framework also allows global optimization strategies to be applied to the final composed FST network.

- **Developed sub-lexical and supra-lexical linguistic models within the proposed framework.**

  Using the proposed framework, we have developed context-dependent hierarchical linguistic models at both sub-lexical and supra-lexical levels. FSTs have been designed and constructed within the proposed framework to encode the structure and probability constraints provided by the hierarchical linguistic models. At the sub-lexical level, a highly regulated grammar is used to describe the sub-lexical linguistic hierarchy. The grammar is combined with context-dependent probability models to better model context-dependent sub-lexical phenomena, such as phonological variations. At the supra-lexical level, the conditional context selection for context-dependent hierarchical linguistic models is first studied, and the benefits for using such context-dependent probabilities across rule boundaries of a natural language grammar are demonstrated. Then, a shallow parsing grammar is derived directly from the full-fledged natural language grammar to impose consistent linguistic constraints in speech recognition. A two-level shallow parse tree is generated according to the shallow parsing grammar. The top level connects phrase tags and filler words, and the phrase level captures the phrase structure. Additional probability constraints are provided by top-level n-gram probabilities and phrase-level context-dependent probabilities.

- **Studied the feasibility and effectiveness of the proposed framework.**

  Through the development of the sub-lexical and supra-lexical models mentioned above, we have studied empirically the feasibility and effectiveness of integrating hierarchical linguistic knowledge into speech recognition using the proposed framework. We have found that hierarchical sub-lexical linguistic modeling is effective in providing generic subword structure and probability constraints for unknown words, which help the recognizer correctly identify them. Together with unknown word support from natural language understanding, a conversational interface would be able to deal with unknown words better, and can possibly incorporate them into the active recognition vocabulary on-the-fly. At the supra-lexical level, experimental results have shown that the shallow parsing model built within the proposed layered FST framework with top-level n-gram

probabilities and phrase-level context-dependent rule-start probabilities is able to reduce recognition errors, compared to a class n-gram model of the same order. However, we have also found that its application can be limited by the complexity of the composed FSTs. This suggests that with a much more complex grammar at the supra-lexical level, a proper tradeoff between tight knowledge integration and system complexity becomes more important.

- **Proposed a dynamic reliability modeling approach to integrate acoustic model reliability knowledge.**

  Another important aspect of achieving more accurate and robust speech recognition is the integration of *acoustic* knowledge. Typically, along a recognizer's search path, some acoustic units are modeled more reliably than others, due to differences in their acoustic-phonetic features and many other factors. We have presented a dynamic reliability scoring scheme which can help adjust partial path scores while the recognizer searches through the composed lexical and acoustic-phonetic network. The reliability models are trained on acoustic scores of a correct arc and its immediate competing arcs extending the current partial path. During recognition, if, according to the trained reliability models, an arc can be more easily distinguished from the competing alternatives, that arc is then more likely to be in the right path, and the partial path score can be adjusted accordingly to reflect such acoustic model reliability information. We have applied this reliability scoring scheme in two weather information domains. The first one is the JUPITER system in English, and the second one is the PANDA system in Mandarin Chinese. We have demonstrated the effectiveness of the dynamic reliability modeling approach in both cases.

We want to emphasize that Chung [18, 19] has proposed a three-stage recognition architecture, which also tries apply sub-lexical linguistic knowledge early in speech recognition to accommodate multi-domain systems with flexible vocabulary. Similar to our sub-lexical modeling approach, The ANGIE sub-lexical model is represented by an weighted FST. However, the FST is constructed by enumerating all parsing columns occurring in training, then connecting them using precomputed ANGIE column bi-gram probabilities. Such an approach is able to capture a large portion of the sub-lexical probability space when phonetic observa-

tions are *well supported* by training data. In contrast, our layered FST sub-lexical modeling approach supports the *full* probability space without limiting the generalization power of probabilistic sub-lexical modeling on previously unseen words, especially when sub-lexical parse trees contain columns not connected or not seen in the training data. Furthermore, it provides a layered view of constructing the hierarchical probability models. This would facilitate choosing a suitable probability model for each sub-lexical layer.

For the rest of this section, we briefly recapitulate our methodologies and results for the main chapters in this thesis.

7.1.1 Sub-lexical Linguistic Modeling

In Chapter 2, we have discussed sub-lexical linguistic modeling in speech recognition. In order to increase flexibility and improve sharing of training data, acoustic models in most modern speech recognition systems are based on units smaller than words. However, high-level linguistic modeling and natural language processing in speech-based conversational interfaces typically operate at the word level. Sub-lexical linguistic modeling bridges the gap between sub-word units and words. It uses sub-word linguistic knowledge to model the construction of words from sub-lexical units.

Typical approaches of modeling low-level phonemic and phonetic linguistic phenomena include pronunciation networks and detailed acoustic models. However, they are not able to support higher-level sub-lexical linguistic knowledge, such as morphology and syllabification. Further research has revealed a more comprehensive picture of sub-word structure. We have introduced the sub-lexical linguistic hierarchy, and outlined the formalization of hierarchical sub-lexical knowledge with CFGs. We have also discussed the motivations and procedures for augmenting CFGs with probability models, including the rule-production probabilities and context-dependent probabilities based on a column-column substructure.

We have also introduced the finite-state techniques used in speech recognition, and presented the possibility of modeling sub-lexical linguistic knowledge using FSTs. Our main objective for sub-lexical modeling is to establish an FST-based framework capable of representing context-dependent hierarchical sub-lexical knowledge, and use it to integrate such knowledge with speech recognition.

7.1.2 Integration of Sub-lexical Models with Speech Recognition

In Chapter 3, we have emphasized our efforts to model hierarchical sub-lexical linguistic knowledge using an FST framework. We begin by exploring FST-based hierarchical models with rule production probabilities at the phoneme level, and compare it with some other approaches such as phoneme networks and phoneme n-gram models. The rule production probabilities are examined first because it is straightforward to integrate such probabilities into RTNs, and then construct an FST-based sub-lexical model from the RTNs. Such a model also allows us to examine the effectiveness of probabilistic structural constraints above the phoneme level within the CFG formalism.

We have shown that the generic sub-word structural constraints provided by hierarchical sub-lexical modeling above the phoneme level are effective, particularly for previously unseen words, because linguistically motivated structure constraints are not restricted to a particular vocabulary. Compared to some other models with the ability to support unseen words, such as the phoneme network with filler models and the phoneme bi-gram model, the hierarchical model is able to provide relatively tight constraints for phoneme sequences from unknown words. However, since low level phonological phenomena are generally context-dependent, it is not likely that sufficient local constraints at the *phone* level can be offered by the rule production probabilities. We have shown that it is necessary to use context-dependent probability models to augment the sub-lexical CFG, especially at low levels of the sub-lexical linguistic hierarchy. With the ANGIE context-dependent probabilities, the *phone* level hierarchical model reduces perplexity substantially compared to the model using the context-independent rule production probabilities.

One main difficulty in incorporating context-dependent probabilities is that they can not be directly encoded by weighted RTNs, because the conditional contexts are beyond the current rule boundary. In order to encode the full sub-lexical hierarchy and the context-dependent probabilities, we have proposed an FST-based sub-lexical modeling framework using a layered approach, where context-dependent probabilities at different layers of the sub-lexical hierarchy are encoded by separate weighted FSTs. A sub-lexical parsing RTN is composed with the layer-specific FSTs to establish the complete sub-lexical model. The FST-based sub-lexical model can be seamlessly integrated with speech recognition in a uni-

fied FST architecture. Such a tight integration interface helps impose sub-lexical linguistic constraints early in the recognition search. With the ANGIE probability model, the layered framework is able to represent the full ANGIE sub-lexical probability space with little sacrificing of its generalization capability. Moreover, the probability models for each layer are not restricted to the ANGIE tri-gram and phone advancement probabilities. With different applications and training data availability, they can be changed to suit different situations. The flexibility to accommodate different probability models is another important benefit offered by the proposed framework.

Our recognition experiments have shown successful support for previously unseen words using context-dependent hierarchical sub-lexical models. We have also demonstrated the feasibility of implementing such models within the proposed layered framework. One further advantage of modeling unseen words with hierarchical linguistic models is that the complete sub-lexical linguistic analysis is available for subsequent processing. For example, such linguistic information can be used to propose the spelling of unknown words, and dynamically incorporate them into the recognition vocabulary.

7.1.3 Supra-lexical Linguistic Modeling

Supra-lexical linguistic knowledge is used in both the speech recognition and the natural language understanding components of a speech understanding system. In speech recognition, the fundamental task of supra-lexical linguistic modeling is to use the *a priori* linguistic knowledge of the language to guide the recognition search. Note that the linguistic knowledge applied in natural language understanding can also benefit speech recognition. In Chapter 4, we have discussed the use of supra-lexical linguistic knowledge in both speech recognition and language understanding components, and proposed an approach to automatically select the conditional context in context-dependent hierarchical supra-lexical models.

In speech recognition, the widely used supra-lexical linguistic model is the n-gram language model. It is able to provide local probability constraints with the flexibility of accepting arbitrary word sequences, and offers satisfactory performance in many cases. Several n-gram variants are introduced to improve the basic word n-gram model. For example, the

class n-gram model can reduce the number of model parameters, which helps obtain a relatively robust model with limited training data. Predictive clustering further introduces class context-dependency while predicting the next word given the class it belongs to. Since the basic n-gram model mainly captures local word connection constraints, structured language models have been introduced to model sentence structure constituents and long-distance constraints with the use of formal grammars. Such grammars are typically syntax-oriented, intended to capture the syntactic structure of a sentence.

The natural language understanding component also makes heavy use of supra-lexical linguistic knowledge. Typically natural language grammar rules are used to formalize the syntactic and semantic knowledge, and probability models can be imposed on the grammars. Such probability models provide additional probability constraints, and offer a quantitative assessment of how well the input word sequence complies with the *a priori* linguistic knowledge. The basic approach to augmenting the natural language CFGs with probabilities is to use stochastic CFGs (SCFGs). However, SCFGs are not able to model the context-dependency beyond the rule boundaries. We have discussed context-dependent probability models, which can account for categorical contexts in a parse tree beyond the current derivation; thus, stronger probability constraints can be imposed. The central issue in laying out a context-dependent probability model with CFGs is to choose the conditional context.

We have proposed an approach that automatically selects effective conditional contexts used in hierarchical context-dependent linguistic modeling, based on the TINA probability model. It has been shown that the context-dependent rule-start probabilities are clearly more effective than the generic rule-start probabilities. The automatic context selection approach is able to provide further improvements over the original context-dependent model with fixed left sibling context nodes. After studying the statistics on the left context chosen, we have explored one simplified scheme, where the left parent is used as the conditional context instead of the left sibling. Experimental results show that using the left parent results in a lower perplexity compared to the original model. The perplexity improvement is moderate compared to the original left sibling context used in TINA, which suggests that the TINA left sibling context is also effective.

7.1.4 Integration of Supra-lexical models with Speech Recognition

In Chapter 5, we have studied the integration of hierarchical supra-lexical linguistic models with speech recognition, and proposed a shallow parsing approach for supra-lexical linguistic modeling. Shallow parsing can help balance sufficient generality and tight phrase structure constraints, which is particularly important in conversational interfaces where spontaneous speech has to be handled. The shallow parsing grammar is directly derived from the full natural language understanding grammar, and is augmented with top-level n-gram probabilities and context-dependent phrase-level probabilities. Such an approach can help impose recognition constraints consistent with natural language understanding.

The proposed context-dependent hierarchical shallow parsing model is constructed within the same layered FST framework as has been used for sub-lexical hierarchical linguistic modeling. A shallow parsing RTN, built from the shallow parsing grammar, outputs a tagged parse string representing the shallow parse tree. It also encodes top-level n-gram probabilities, phrase-level generic rule-start probabilities, and phrase-level rule-internal probabilities. A separate FST is used to model phrase-level context-dependent rule-start probabilities, and is composed with the parsing RTN to build the complete supra-lexical linguistic model. Such a unified FST framework is able to seamlessly integrate the shallow parsing based supra-lexical linguistic model into speech recognition. It also has the potential of providing early feedback from natural language understanding in the speech recognition search.

Our speech recognition experiments show that the proposed context-dependent shallow parsing model with top-level n-gram probabilities and phrase-level context-dependent probabilities is able to reduce recognition error, compared to a regular class n-gram model with the same order. However, the final composed FST representing the speech recognition search space can be significantly larger than the regular class n-gram model. With a higher order top-level n-gram, pre-composition and optimization are restricted by the computational resources available. A compromise between tight knowledge integration and system complexity may be necessary while applying such an FST-based context-dependent supra-lexical linguistic model.

7.1.5 Dynamic Reliability Modeling in Speech Recognition

In Chapter 6, we have proposed a dynamic reliability modeling approach that integrates acoustic model reliability information directly in the recognition search. Reliability models are used to formalize the notion that, along the recognition search path, some acoustic units may be modeled and recognized more reliably than others. Confidence scoring and rejection methods also address this problem; however, confidence scores are usually obtained after the completion of the recognition search, and thus do not alter the search strategy. In contrast, our reliability models are trained to characterize typical acoustic scores of an arc in the *correct* path and the *competing* paths from the recognition search network, and are used to provide acoustic model reliability information on-the-fly during the recognition search. Such a dynamic integration approach can help early recovery from recognition errors. The recognizer's acoustic models and reliability models are trained and evaluated in two weather information domains: JUPITER in English with relatively large amount of training data, and PANDA in Mandarin Chinese with limited training data. For the JUPITER domain, both di-phone and segment based acoustic models are used. The reliability models are built on the combined di-phone and segment acoustic scores. For the PANDA domain, due to insufficient data for training di-phone models, only segment based acoustic models are used, and therefore the reliability models are built on segment acoustic scores. We have shown that dynamic reliability modeling is able to improve the recognition accuracy in both the JUPITER and PANDA domain. Furthermore, it has been demonstrated that iterative training is helpful to improve the acoustic models and the reliability models.

7.2 Future Directions

It is an appealing idea to tightly integrate different knowledge sources in speech recognition using a unified framework. Much effort has been put into research related to such a seamless knowledge integration scheme [74, 110, 99, 57, 19], and the results have demonstrated that early and tight integration of knowledge sources is able to benefit speech recognition. However, we believe that, with the increase in complexity of application domains and corresponding speech recognition systems, such a tight integration scheme may lead to dra-

matically increased cost in creating the integrated search space. With limited computational resources, It becomes necessary to balance accuracy and efficiency.

One possible solution is to construct a recognition framework where the tightness of knowledge integration can be controlled and optimized. This strategy should allow easy adjustment of how and when the different levels of knowledge are incorporated. For simpler domains and recognition systems, knowledge sources can be combined tightly all at once. For more complex systems, the stages where different knowledge sources are introduced can be changed to achieve relatively tight integration with manageable complexity for system construction. Finite-state transducers can still be used towards such a flexible knowledge integration architecture. The basic strategy can be outlined as follows. Similar to our layered FST framework, different knowledge sources at different levels can first be represented using separate FSTs. After that, the intermediate FSTs are systematically pruned and composed until the final FST representing the search space is constructed. In extreme cases, where aggressive pruning, or no pruning, is performed at each FST composition step, a system with relatively loose, or tight, knowledge integration can be established, respectively. Then the central issue becomes the development of the optimal composition and pruning strategy according to the complexity of different tasks and the availability of computational resources. This is a possible future direction for designing general knowledge formalization and integration frameworks for speech recognition.

Next, we summarize some specific future studies related to our efforts in modeling and integrating sub-lexical linguistic knowledge, supra-lexical linguistic knowledge, and dynamic reliability knowledge.

- **Modeling and Integrating Sub-lexical Linguistic Knowledge**

  Hierarchical sub-word linguistic knowledge is able to provide sub-word structural constraints for speech recognition. We have shown in section 3.3.1 that the structural constraints are effective with rule production probability models above the phoneme level, and such probabilities can be conveniently encoded using weighted RTNs. However, when the sub-lexical hierarchy is extended to the phone level, the highly context-dependent phonological variations can not be well modeled by the context-independent rule production probabilities. Seneff and Wang [102] have proposed an approach to model phonolog-

ical rules with the ANGIE system, where a probability model is designed to capture only phonemic to phonetic predictions in context. It is possible to combine such a phone level probability model with the higher level rule-production probability models within our proposed FST framework. Compared to the layered probability FST approach, where context-dependent probabilities are used at all sub-word layers, such an combined approach may produce a more compact FST search space, since a single weighted RTN is able to integrate the probability models above the phoneme level.

Another important application of sub-lexical linguistic modeling is to provide sub-word linguistic support for words not in the recognizer's vocabulary. Previously unseen words are a major source of misrecognition, especially in conversational interfaces, where spontaneous speech needs to be handled. Since our hierarchical sub-lexical linguistic models are able to provide detailed linguistic analysis according to the sub-lexical CFG, such information can be used to help hypothesize the spelling of the new word, and dynamically incorporate them in the recognition vocabulary. A detailed study of flexible vocabulary recognition using sub-lexical linguistic knowledge is given by Chung in [19]. It is potentially beneficial to construct flexible vocabulary recognition systems within the proposed FST-based framework.

Previously unseen words can also be handled by applying statistical methods to model sub-word probability constraints. Bazzi [10] has focused his research using this approach. It is possible to combine the linguistically motivated approach and the data-driven approach to achieve more robust unseen word modeling.

- **Modeling and Integrating Supra-lexical Linguistic Knowledge**

  As we have mentioned before, the main problem we have encountered in supra-lexical linguistic modeling using the layered FST-based framework is the complexity of the FSTs. We have used a shallow parsing approach, which simplifies the original full natural language grammar. However, the complexity of FSTs is still a major concern. One possible strategy is to use an incremental approach similar to the FST-based incremental n-gram model [30], where the complete supra-lexical model is factored into two FSTs. The first one can be statically composed and optimized, and the second one is composed on-the-

fly. This approach is able to include part of the supra-lexical linguistic model FST into early pre-composition and optimization, while keeping the complete supra-lexical model through incremental dynamic composition.

- **Dynamic Reliability Modeling**

  Currently the reliability models are established based on the candidate arcs directly connected to the current partial path. The reliability scores are only used to help the recognizer choose the best arc from its *immediate* competitors. Ideally, we would like to provide reliability measurements based on all the future arcs extending the current path. Since this may be highly computationally expensive, a possible compromise is to consider extending the current partial path within a certain interval. This would help reduce the bias of only considering the immediate future context, thus giving better reliability adjustments.

  We have also noticed that, with a given recognition lexicon, it is usually not necessary to accurately recognize every acoustic unit in the word to get correct recognition results. It is possible to use only several reliable pieces of the word to distinguish it from other words. Therefore, it can be beneficial to obtain such reliable pieces with the guidance of the reliability measurements, and change the lexical access scheme from precise matching to reliable-island matching. Such an island-driven lexical access approach has the advantage of using reliably modeled and less variant phones to characterize words with complex pronunciation variations. It can also help address the non-native speech [64] problem.

Bibliography

[1] T. Akiba and K. Itou, "A structured statistical language model conditioned by arbitrarily abstracted grammatical categories based on GLR parsing," in *Proc. EuroSpeech'01*, Aalborg, Denmark, pp. 705–708, 2001.

[2] L. Bahl, J. Baker, P. Cohen, F. Jelinek, B. Lewis, and R. Mercer, "Recognition of a continuously read natural corpus," in *Proc. ICASSP'80*, Denver, CO, 1980.

[3] L. Bahl, J. Bellegarda, P. De Souza, P. Gopalakrishnan, D. Nahamoo, and M. Picheny, "Multonic Markov word models for large vocabulary continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 3, pp. 334–344, 1993.

[4] L. Bahl, P. Brown, P. De Souza, R. Mercer, and M. Picheny, "Acoustic Markov models used in the TANGORA speech recognition system," in *Proc. ICASSP'88*, New York, NY, 1988.

[5] L. Bahl, P. Brown, P. De Souza, R. Mercer, and M. Picheny, "A method for the construction of acoustic Markov models for words," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 4, pp. 443–452, 1993.

[6] L. Bahl, P. De Souza, P. Gopalakrishnan, and M. Picheny, "Decision trees for phonological rules in continuous speech," in *Proc. ICASSP'91*, Toronto, Canada, pp. 185–188, 1991.

[7] J. Baker, "The DRAGON system: An overivew," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 24–29, 1975.

[8] J. Baker, "Trainable grammars for speech recogntion," in *Proc. Spring Conference of the Acoustic Society of America*, Cambridge, MA, pp. 547–550, 1979.

[9] L. Baptist and S. Seneff, "GENESIS-II: A versatile system for language generation in conversational system applications," in *Proc. ICSLP'00*, Beijing, China, 2000.

[10] I. Bazzi and J. Glass, "Modeling out-of-vocabulary words for robust speech recognition," in *Proc. ICSLP'00*, Beijing, China, 2000.

[11] E. Black, F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos, "Towards history-based grammars: Using richer models for probabilistic parsing," in *Proc. ACL'93*, Columbus, OH, pp. 31–37, 1993.

[12] R. Bod, "Enriching linguistics with statistics: Performance models of natural language," in *ILLC Dissertation Series*, Amsterdam, 1995.

[13] S. Boisen, "The BBN spoken language system," in *Proc. Speech and Natural Language Workshop*, Philadelphia, PA, pp. 106–111, 1989.

[14] T. Booth, "Probabilistic representation of formal languages," in *IEEE Conference Record of the 1969 Tenth Annual Symposium on Switching and Automata Theory*, pp. 74–81, 1969.

[15] T. Briscoe and J. Carroll, "Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars," *Computational Linguistics*, vol. 19, no. 1, pp. 25–59, 1993.

[16] C. Chelba and F. Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, no. 4, pp. 283–332, 2000.

[17] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz, "BYBLOS: The BBN continuous speech recognition system," in *Proc. ICASSP'87*, Dallas, TX, 1987.

[18] G. Chung, "A three-stage solution for flexible vocabulary speech understanding," in *Proc. ICSLP'00*, Beijing, China, 2000.

[19] G. Chung, *Towards Multi-domain Speech Understanding with Flexible and Dynamic Vocabulary*. Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[20] K. Church, *Phrase-Structure Parsing: A Method for Taking Advantage of Allophonic Constraints*. Ph.D. thesis, Massachusetts Institute of Technology, 1983.

[21] K. Church, *Phonological Parsing in Speech Recogntion*. Massachusetts, USA: Kluwer Academic Publishers, 1987.

[22] K. Church, "A stochastic parts program and noun phrase parser fro unrestricted text," in *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, pp. 136–143, 1988.

[23] G. Clements and S. Keyser, *CV Phonology: A Generative Theory of the Syllable*. Cambridge, MA, USA: The MIT Press, 1983.

[24] P. Cohen and R. Mercer, "The phonological component of an automatic speech recognition system," in *Proc. IEEE Symposium on Speech Recognition*, Pittsburgh, PA, USA, pp. 177–187, 1974.

[25] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, *Survey of the State of the Art in Human Language Techonology*. Cambridge, UK: Cambridge University Press, 1997.

[26] R. Cole, R. Stern, M. Phillips, S. Brill, A. Pilant, and P. Specker, "Feature-based speaker-independent recognition of isolated English letters," in *Proc. ICASSP'83*,, pp. 731–734, 1983.

[27] R. De Mori and M. Galler, "The use of syllable phonotactics for word hypothesization," in *Proc. ICASSP'96*, Atlanta, Georgia, USA, pp. 877–880, 1996.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[29] L. Deng, J. Wu, and H. Sameti, "Improved speech modeling and recognition using multi-dimensional articulatory states as primitive speech units," in *Proc. ICASSP'95*, Detroit, Michigan, USA, pp. 385–388, 1995.

[30] H. Dolfing and I. Hetherington, "Incremental language models for speech recognition using finite-state transducers," in *Proc. IEEE Automatic Speech Recogntion and Understanding Workshop*, Madonna di Campiglio, Italy, 2001.

[31] A. Fisher, "Practical parsing of generalized phrase structure grammar," *Computational Linguistics*, vol. 15, no. 3, pp. 139–148, 1989.

[32] E. Fudge, "Syllables," *Journal of Linguistics*, vol. 5, pp. 253–285, 1969.

[33] L. Galescu, E. Ringger, and J. Allen, "Rapid language model development for new task domains," in *Proc. International Conference on Language Resources and Evaluation*, Granada, Spain, 1998.

[34] G. Gazdar, E. Klein, G. Pullum, and I. Sag, *Generalized Phrase Structure Grammars*. Cambridge, Massachusetts, USA: Harvard University Press, 1985.

[35] P. Geutner, "Introducing linguistic constraints into statistical language modeling," in *Proc. ICSLP'96*, Philadelphia, PA, USA, 1996.

[36] J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. ICSLP'96*, Philadelphia, PA, USA, pp. 2277–2280, 1996.

[37] J. Glass, T. Hazen, and I. Hetherington, "Real-time telephone-based speech recognition in the JUPITER domain," in *Proc. ICASSP'99*, Phoenix, AZ, USA, 1999.

[38] J. Glass and E. Weinstein, "SPEECHBUILDER: Facilitating spoken dialogue systems development," in *Proc. EuroSpeech'01*, Aalborg, Denmark, 2001.

[39] D. Goddeau, *An LR parser based probabilistic language model for spoken language systems*. Ph.D. thesis, Massachusetts Institute of Technology, 1993.

[40] J. Goodman and J. Gao, "Language model size reduction by pruning and clustering," in *Proc. ICSLP'00*, Beijing, China, 2000.

[41] T. Hazen, I. Hetherington, and A. Park, "FST-based recognition techniques for multilingual and multi-domain spontaneous speech," in *Proc. EuroSpeech'01*, Aalborg, Denmark, 2001.

[42] I. Hetherington, "An efficient implementation of phonological rules using finite-state transducers," in *Proc. EuroSpeech'01*, Aalborg, Denmark, 2001.

[43] D. Hindle, "Acquiring disambiguation rules from text," in *Proc. ACL'89*, Vancouver, Canada, pp. 118–125, 1989.

[44] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[45] F. Jelinek, L. Bahl, and R. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," *IEEE Trans. on Information Theory*, vol. 21, no. 3, pp. 250–256, 1975.

[46] T. Jitsuhiro, H. Yamamoto, S. Yamada, and Y. Sagisaka, "New language models using phrase structures extracted from parse trees," in *Proc. EuroSpeech'01*, Aalborg, Denmark, pp. 697–700, 2001.

[47] D. Kahn, *Syllable-based Generalizations in English Phonology.* Ph.D. thesis, Massachusetts Institute of Technology, 1976.

[48] S. Kamppari and T. Hazen, "Word and phone level acoustic confidence scoring," in *Proc. ICASSP'00*, Istanbul, Turkey, 2000.

[49] R. Kaplan and M. Kay, "Regular models of phonological rule systems," *Computational Linguistics*, vol. 20, no. 3, pp. 331–378, 1994.

[50] L. Karttunen, R. Kaplan, and A. Zaenen, "Two-level morphology with composition," in *Proc. COLING'92*, Nantes, France, pp. 141–148, 1992.

[51] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 35, 1987.

[52] D. Klatt, *Trends in Speech Recognition, Chapter 25.* Prentice-Hall, 1980.

[53] K. Koskenniemi, *Two-level Morphology: A General Computational Model for Word Form Recognition and Production.* Publication No. 11, Department of General Linguistics, University of Helsinki, 1983.

[54] S. Kullback and A. Leibler, "On information and sufficiency," *Annals of Math Statistics*, vol. 22, pp. 79–86, 1951.

[55] J. Lafferty and D. Sleator, "Grammatical trigrams: A probabilistic model of link grammar," in *Proc. 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, 1992.

[56] R. Lau, *Subword Lexical Modelling for Speech Recognition.* Ph.D. thesis, Massachusetts Institute of Technology, 1998.

[57] R. Lau and S. Seneff, "A unified framework for sublexical and linguistic modelling supporting flexible vocabulary speech understanding," in *Proc. ICSLP'98*, Sydney, Australia, 1998.

[58] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Improved acoustic modeling for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 6, no. 2, pp. 103–127, 1992.

[59] K. Lee, *Automatic Speech Recognition*. Massachusetts, USA: Kluwer Academic Publishers, 1983.

[60] K. Lee, *Large-vocabulary speaker-independent continuous speech recognition: The* SPHINX *system*. Ph.D. thesis, Carnegie-Mellon University, 1988.

[61] M. Lennig, D. Sharp, P. Kenny, V. Gupta, and K. Precoda, "Flexible vocabulary recognition of speech," in *Proc. ICSLP'92*, Banff, Canada, pp. 93–96, 1992.

[62] V. Lesser, R. Fennel, L. Erman, and D. Reddy, "Organization of the HEARSAY II speech understanding system," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 11–24, 1975.

[63] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. ICASSP'87*, pp. 705–708, 1987.

[64] K. Livescu and J. Glass, "Lexical modeling of non-native speech for automatic speech recognition," in *Proc. ICASSP'00*, Istanbul, Turkey, 2000.

[65] M. McCandless, *Automatic acquisition of language models for speech recognition*. Master's thesis, Massachusetts Institute of Technology, 1994.

[66] H. Meng, *Phonological Parsing for Bi-directional Letter-to-Sound / Sound-to-Letter Generation*. Ph.D. thesis, Massachusetts Institute of Technology, 1995.

[67] H. Meng, S. Hunnicutt, S. Seneff, and V. Zue, "Reversible letter-to-sound / sound-to-letter generation based on parsing word morphology," *Speech Communication*, vol. 18, pp. 47–63, 1996.

[68] M. Mohri, "On some applications of finite-state automata theory to natural language processing," *Natural Language Engineering*, vol. 2, pp. 1–20, 1996.

[69] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, no. 2, pp. 269–311, 1997.

[70] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducer in speech recognition," in *Proc. ISCA ASR'00*, Paris, France, 2000.

[71] M. Mohri, F. Pereira, and M. Riley, "The design principles of a weighted finite-state transducer library," *Theoretical Computer Science*, pp. 17–32, January 2000.

[72] M. Mohri and M. Riley, "Network optimization for large vocabulary speech recognition," *Speech Communication*, vol. 25, 1998.

[73] M. Mohri and R. Sproat, "An efficient compiler for weighted rewrite rules," in *Proc. ACL'96*, Santa Cruz, USA, pp. 231–238, 1996.

[74] R. Moore, "Integration of speech with natural language understanding," in *Voice Communication between Humans and Machines* (D. Roe and J. Wilpon, eds.), pp. 254–279, National Academy Press, 1994.

[75] R. Moore, F. Pereira, and H. Murveit, "Integrating speech and natural-language processing," in *Proc. Speech and Natural Language Workshop*, Philadelphia, PA, pp. 243–347, 1989.

[76] S. Mori, M. Nishimura, and N. Itoh, "Improvement of a structured language model: Arbori-context tree," in *Proc. EuroSpeech'01*, Aalborg, Denmark, 2001.

[77] X. Mou and V. Zue, "The use of dynamic reliability scoring in speech recognition," in *Proc. ICSLP'00*, Beijing, China, 2000.

[78] H. Murveit and R. Moore, "Integrating natural language constraints into HMM-based speech recognition," in *Proc. ICASSP'90*, Albuquerque, NM, USA, pp. 573–576, 1990.

[79] M. Nederhof, "Practical experiments with regular approximation of context-free languages," *Computational Linguistics*, vol. 26, no. 1, pp. 17–44, 2000.

[80] M. Nishimura and K. Toshioka, "Hmm-based speech recognition using multi-dimensional multi-labelling," in *Proc. ICASSP'87*, pp. 1163–1166, 1987.

[81] K. Paliwal, "Lexicon-building methods for an acoustic sub-word based speech recognizer," in *Proc. ICASSP'90*, Albuquerque, NM, USA, pp. 729–732, 1990.

[82] C. Pao, P. Schmid, and J. Glass, "Confidence scoring for speech understanding systems," in *Proc. ICSLP'98*, Sydney, Australia, 1998.

[83] M. Phillips, J. Glass, and V. Zue, "Modeling context dependency in acoustic phonetic and lexical representations," in *Proc. Speech and Natural Language Workshop*, Asilomar, California, pp. 71–76, 1991.

[84] C. Pollard and I. Sag, *Head-Driven Phrase Structure Grammar*. Chicago, USA: University of Chicago Press, 1994.

[85] P. Price, "Evaluation of spoken language systems: the ATIS domain," in *Proc.* DARPA *Speech and Natural Language Workshop*, Philadelphia, PA, pp. 91–95, 1990.

[86] L. Rabiner, B. Juang, S. Levinson, and M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Technical Journal*, vol. 63, no. 7, pp. 1245–1260, 1985.

[87] P. Resnik, "Probabilistic tree-adjoining grammar as a framework for statistical natural language processing," in *Proc. COLING'92*, Nantes, France, pp. 418–424, 1992.

[88] K. Ries, F. Buo, and A. Waibel, "Class phrase models for language modeling," in *Proc. ICSLP'96*, Philadelphia, PA, USA, 1996.

[89] E. Roche and Y. Schabes, *Finite-State Language Processing*. Cambridge, MA, USA: The MIT Press, 1997.

[90] J. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, and M. Siu, "Phonetic training and language modeling for word spotting," in *Proc. ICASSP'93*, Minneapolis, MN, USA, pp. 459–462, 1993.

[91] R. Rose, "Definition of subword acoustic units for word spotting," in *Proc. EuroSpeech'93*, Berlin, Germany, pp. 1049–1052, 1993.

[92] A. Rudnicky, L. Baumeister, K. DeGraaf, and E. Lehmann, "The lexical access component of the CMU continuous speech recognition system," in *Proc. ICASSP'87*, 1987.

[93] R. Schwartz, Y. Chow, S. Roucos, M. Krasner, and J. Makhoul, "Improved hidden Markov modeling of phonemes for continuous speech recognition," in *Proc. ICASSP'84*, San Diego, CA, 1984.

[94] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, vol. 18, no. 1, pp. 61–86, 1992.

[95] S. Seneff, "The use of linguistic hierarchies in speech recognition," in *Proc. ICSLP'98*, Sydney, Australia, 1998.

[96] S. Seneff, C. Chuu, and D. Cyphers, "Orion: From on-line interaction to off-line delegation," in *Proc. ICSLP'00*, Beijing, China, 2000.

[97] S. Seneff, R. Lau, and H. Meng, "ANGIE: A new framework for speech analysis based on morpho-phonological modelling," in *Proc. ICSLP'96*, Philadelphia, PA, USA, 1996.

[98] S. Seneff, R. Lau, and J. Polifroni, "Organization, communication and control in the GALAXY-II conversational system," in *Proc. EuroSpeech'99*, Budapest, Hungary, 1999.

[99] S. Seneff, M. McCandless, and V. Zue, "Integrating natural language into the word graph search for simultaneous speech recogntion and understanding," in *Proc. EuroSpeech'95*, Madrid, Spain, 1995.

[100] S. Seneff and J. Polifroni, "Dialogue management in the MERCURY flight reservation system," in *Satellite Dialogue Workshop, ANLP-NAACL*, Seattle, WA, USA, 2000.

[101] S. Seneff and J. Polifroni, "Formal and natural language generation in the MERCURY conversational system," in *Proc. ICSLP'00*, Beijing, China, 2000.

[102] S. Seneff and C. Wang, "Modeling phonological rules through linguistic hierarchies," in *Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology Workshop (submitted)*, Estes Park, CO, 2002.

[103] S. Shieber, *An Introduction to Unification-Based Approaches to Grammars*. Stanford, California, USA: Center for the Study of Language and Information, Leland Stanford Junior University, 1986.

[104] J. Shoup, "Phonological aspects of speech recognition," in *Trends in Speech Recognition* (W. Lea, ed.), pp. 125–138, Englewood Cliffs, NJ: Prentice Hall, 1980.

[105] S. Siegel and N. Castellan, *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw Hill, 1988.

[106] M. Sipser, *Introduction to the Theory of Computation*. Massachusetts, USA: PWS Publishing Company, 1997.

[107] A. Suchato, *Framework for joint recognition of pronounced and spelled proper names*. Master's thesis, Massachusetts Institute of Technology, 2000.

[108] C. Wang, S. Cyphers, X. Mou, J. Polifroni, S. Seneff, J. Yi, and V. Zue, "MUXING: A telephone-access mandarin conversational system," in *Proc. ICSLP'00*, Beijing, China, 2000.

[109] W. Ward, "The CMU air travel information service: Understanding spontaneous speech," in *Proc. DARPA Speech and Natural Language Workshop*, pp. 127–129, 1990.

[110] W. Ward and S. Issar, "Integrating semantic constraints into the SPHINX-II recognition search," in *Proc. ICASSP'94*, Adelaide, Australia, pp. 2017–2019, 1994.

[111] A. Wendemuth, G. Rose, and J. Dolfing, "Advances in confidence measures for large vocabulary," in *Proc. ICASSP'99*, Phoenix, USA, 1999.

[112] K. Wilson, *The Columbia Guide to Standard American English*. New York, NY: Columbia University Press, 1993.

[113] V. Zue and J. Glass, "Conversational interfaces: Advances and challenges," *Proc. IEEE Special Issue on Spoken Language Processing*, vol. 88, no. 8, pp. 1166–1180, 2000.

[114] V. Zue, J. Glass, D. Goodline, 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*, Toronto, Canada, pp. 713–716, 1991.

[115] V. Zue, J. Glass, D. Goodline, M. Phillips, and S. Seneff, "The SUMMIT speech recognition system: Phonological modelling and lexical access," in *Proc. ICASSP'90*, Alburquerque, NM, USA, pp. 49–52, 1990.

[116] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and I. Hetherington, "JUPITER: A telephone-based conversational interface for weather information," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.