Discovering Linguistic Structures in Speech: Models and Applications

by

Chia-ying (Jackie) Lee

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

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

September 2014

Signature of Author: _____

Department of Electrical Engineering and Computer Science
July 22, 2014

Certified by: _____

James R. Glass, Senior Research Scientist
Thesis Supervisor

Accepted by: _____

Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Discovering Linguistic Structures in Speech: Models and Applications

by Chia-ying (Jackie) Lee

Submitted to the Department of Electrical Engineering
and Computer Science on July 22, 2014
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The ability to infer linguistic structures from noisy speech streams seems to be an innate human capability. However, reproducing the same ability in machines has remained a challenging task. In this thesis, we address this task, and develop a class of probabilistic models that discover the latent linguistic structures of a language directly from acoustic signals. In particular, we explore a nonparametric Bayesian framework for automatically acquiring a phone-like inventory of a language. In addition, we integrate our phone discovery model with adaptor grammars, a nonparametric Bayesian extension of probabilistic context-free grammars, to induce hierarchical linguistic structures, including sub-word and word-like units, directly from speech signals. When tested on a variety of speech corpora containing different acoustic conditions, domains, and languages, these models consistently demonstrate an ability to learn highly meaningful linguistic structures.

In addition to learning sub-word and word-like units, we apply these models to the problem of one-shot learning tasks for spoken words, and our results confirm the importance of inducing intrinsic speech structures for learning spoken words from just one or a few examples. We also show that by leveraging the linguistic units our models discover, we can automatically infer the hidden coding scheme between the written and spoken forms of a language from a transcribed speech corpus. Learning such a coding scheme enables us to develop a completely data-driven approach to creating a pronunciation dictionary for the basis of phone-based speech recognition. This approach contrasts sharply with the typical method of creating such a dictionary by human experts, which can be a time-consuming and expensive endeavor. Our experiments show that automatically derived lexicons allow us to build speech recognizers that consistently perform closely to supervised speech recognizers, which should enable more rapid development of speech recognition capability for low-resource languages.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist

Acknowledgments

About six years ago, when I watched the most senior student in the group defending his thesis, between me and the podium seemed to be the furthest distance in the world. It is still hard to believe that I am almost done with this journey. It has been a wonderful experience, and I know for sure that I could not have walked through this distance without many people, who have been an important part of my life for the past six years.

First of all, I would like to thank my thesis advisor, Jim Glass, for his constant encouragements and patient guidance. This thesis would not have been possible without many inspiring disussions with Jim. Jim's vision towards unsupervised learning from speech data has motivated me to explore the research topics investigated in this thesis. It's been a really great pleasure working with Jim. I also really appreciate that Jim has always trusted in my ability and given me freedom to be creative.

I would also like to thank my thesis committee members, Regina and Victor. I am truly grateful for Regina's confidence in me. Her encouragements always brought me strength whenever my research was not going well. I am also very thankful that Victor gave me the oppotunity to join the Spoken Language Systems (SLS) group, and was always there to provide help when needed.

The SLS group is where I grew over the past six years. I would like to thank every member in the group for creating a fun and friendly research environment: Jim, Scott, Najim, Stephanie, Victor, Chengjie, Patrick, Marcia, Tuka, Carrie, Ekapol, Jen, Xue, David, Mandy, Ann, Daniel, Michael, Stephen, and Yu. I really appreciate everyone for always giving me feedback on my research ideas, on my paper and thesis drafts, and on my presentations. In particular, I would like to thank Stephen for carefully reading many of my paper drafts and Mandy for proofreading my thesis. Special thanks go to Marcia for helping me with many problems I had. I truly enjoyed listening to Marcia sharing her words of wisdom about life. I would also like to thank Scott for spending an entire afternoon teaching me tricks for working with operating systems, and kindly helping me with technical problems during one of the weekends right before my thesis draft deadline.

I would like to thank my officemates in particular. Over the past six years, I had the pleasure to be officemates with Hung-an, Yaodong, Paul, Yuan, Ekapol, Yu, and Mandy. My officemates

were always the first group of people that I went to if I had any questions. I would like to thank Hung-an, Ekapol, and Yu for helping me with many technical problems for training recognizers. I would like to thank Yaodong for helping me with formulating research problems when I was seeking for a thesis topic. Many thanks go to Yuan for introducing me to 6.804 and Paul Hsu for giving me precious advice for being a grad student. Finally, I really appreciate that Yoong Keok Lee had pointed me to a paper that changed my research direction and led to the series of research problems that I worked on for the thesis.

Over the past six years, I was fortunate to be able to collaborate with many great people. I would like to thank Tim O'Donnell for patiently brainstorming and answering all the questions I had ever since our collaboration started. Tim's broad knowledge in human languages has made a profound impact on the third chapter of this thesis. I would also like to thank my officemate Yu Zhang for helping me with experiments that are reported in Chapter 5. I also really appreciate Brenden Lake's for being interested in my work and taking an intiative to start our collaboration. Brenden showed me how to collaborate seamlessly with people from different fields. I am deeply grateful for all the help that Matthew Johnson has given to me. Matt is one of the most brilliant people I have ever met, and the most special thing about him is that he is always so generous and willing to help. I truly hope that one day we can collaborate on some fun projects again. Finally, I would like to thank Mark Johnson from the Macquarie University for making his implementation for adaptor grammars publicly accessible.

Last but not least, I would like to thank my friends for their generous support and my family for their unconditional love. I am so lucky to have all of you in my life. Thank you all for sharing this wonderful journey with me.

Bibliographic Note

Some of the work presented in this thesis has appeared in previous peer-reviewed publications. The work on phonetic inventory discovery directly from speech data, which is discussed in Chapter 2, was first published in Lee et al. [111]. The application of this model to one-shot learning of spoken words and to word pronunciation induction, presented in Chapter 4 and Chapter 5 respectively, were originally published in Brenden et al. [101] and Lee et al. [112]. Finally, a manuscript tailored for the augumented adaptor grammars for inferring hierarchical linguistic structures from acoustic signals, which is investigated in Chapter 3, is currently under preparation.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Overview

Language is full of structure: sentences are composed of words, words consist of syllables, and syllables contain phonetic units. Understanding the latent structures in language is crucial to the process of human language acquisition [97]. Since infants are born, they continuously receive speech streams from people in their surrounding environment. Without much guidance, infants gradually learn to interpret speech: distinguishing between individual phonemes [98, 48, 118, 31, 172], building phonemic categories specific to their native language [184, 18, 99], segmenting as well as categorizing syllables and words [160, 159, 108, 128, 96], associating meaning with words [36], combining words [21] and recognizing grammars to form sentences [61, 62]. If machines are able to automatically discover these hidden linguistic structures automatically as infants do, then we may be able to create intelligent systems that learn to comprehend human languages autonomously. In addition, given that understanding linguistic structures of a language is the key to building Automatic Speech Recognition (ASR) systems, if the phonetic and lexical structures of a language can be inferred automatically, then we may be able to develop ASR systems in a less supervised or totally unsupervised way. Much work has examined the problem of inferring linguistic structures embedded in *text data*, such as grammar induction. However, possibly prohibited by the problem complexity, little literature has investigated linguistic structure discovery directly from speech.

The goal of this thesis is therefore to devise unsupervised models that infer linguistic structures from *speech data*. Our motivation is both scientific and practical.

- Scientifically, from a cognitive viewpoint, this problem is similar to the early task that infants must deal with when learning their first language. There is abundant evidence showing that infants are capable of discriminating between different phonemes and learning to recognize language-specific phonetic categories at an early age [98, 48, 118, 184, 18, 99].

In addition, by the age of 8 months, infants are shown to be able to detect word boundaries in continuous speech [160]. Even though many computational models have been proposed to capture these early language acquisition behaviors, most of the proposed models are designed for less realistic input data such as truncated speech segments of particular phonemes, manually annotated phone sequences, or phone transcriptions generated by ASR systems [181, 60, 13, 25, 33, 34]. Compared to the sensory speech stream that infants are exposed to, these data are highly processed clean signals. As a result, it is unclear how well these models can perform on noisier input data. Furthermore, because of the mismatch in input data type, it is also not clear how closely these models resemble the human language acquisition process. A computational model that induces linguistic structure of a language directly from the continuous speech stream, therefore, not only makes a breakthrough from previous simplified assumptions on input data type, but also provides a means to more closely capture the human language acquisition process.

- Discovery of linguistic structure from acoustic signals also has many practical values. First, the inferred linguistic knowledge can assist linguists in studying a language. For example, examining the sequence of inferred phonetic units embedded in acoustic signals may help linguists learn phonological properties of the language. Second, the induced linguistic structures, such as phonetic and lexical units, can be utilized to extract important content information about the speech data for applications such as spoken document summarization and information retrieval [114, 116, 66, 27]. For example, by treating the discovered lexical items as *pseudo words*, we can represent a spoken document as a bag of *pseudo words*, which can then be used as input for text-processing systems. Given the vast amount of speech content available online such as lecture recordings[1], news broadcasts, and audio books[2], this application of discovered linguistic structure is particularly useful. Finally, discovering linguistic structures in speech also allows us to develop unsupervised training procedures for ASR systems. For instance, by modeling the set of discovered phonetic units with Hidden Markov Models (HMMs) [153, 87], we can obtain an acoustic model for a language without any transcribed speech data. Furthermore, with some access to the word level transcriptions of the acoustic data, we can infer the mapping scheme between the graphemes and the induced phonetic units, and automatically create a pronunciation lexicon, which is an essential requirement for building an ASR system.

[1] For instance, http://ocw.mit.edu/ (MIT OpenCourseWare).

[2] For example, https://librivox.org/ (LibriVox, free public domain audiobooks).

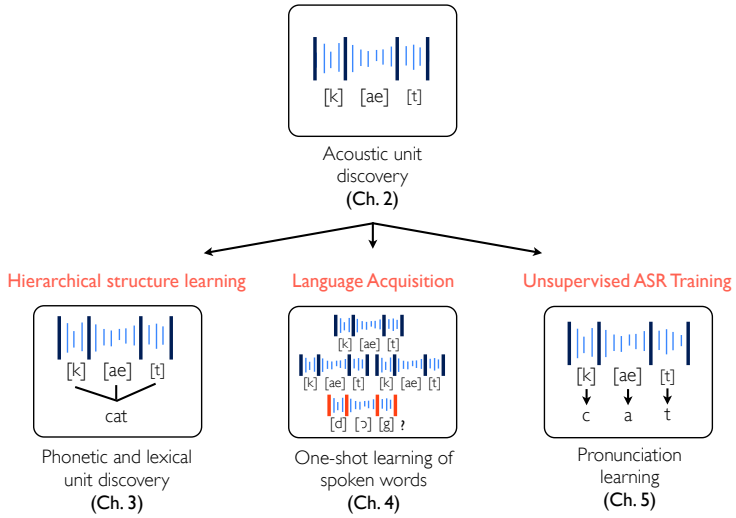


Figure 1.1. The scope of this thesis. This thesis starts with an investigation of acoustic unit discovery from continuous speech, which is then extended to the problem of hierarchical linguistic structure discovery. We furhter develop the idea of unsupervised linguistic structure learning from acoustic signals in two domains, *language acquisition* and *unsupervised ASR training*. Note that except for the task of pronunciation learning, all the phone labels and word transcriptions are shown only for illustration.

We start our investigation with the problem of *unsupervised acoustic phone-like unit discovery*, the problem of finding phone-like units in continuous speech. After successfully discovering phonetic units, we approach an even more difficult task that is joint learning of the phonetic, syllabic and lexical structures of a language directly from continuous speech – the early challenges that infants must tackle to acquire their mother tongue. After developing two models for automatic linguistic structure discovery, we further extend our investigation and apply the models to two domains. The first domain is *language acquisition*. Within this domain, we study the problem of one-shot learning of spoken words – the remarkable human ability to recognize and generalize novel spoken words from only one example. The second domain we focus on is *unsupervised ASR training*, where our objective is to unveil the latent encoding scheme between the writing and the spoken systems of a language, which is essential in converting continuous speech signals to symbolic phoneme representations for ASR systems.

The scope of this thesis is depicted in Fig. 1.1. In the next section, we describe the problem of unsupervised acoustic phone-like unit discovery and briefly discuss our approach. For the sake of brevity, throughout the rest of this thesis, we use the terms *acoustic unit* and *sub-word unit* to represent *acoustic phone-like unit*.

■ 1.2 Unsupervised Acoustic Unit Discovery

Discovering acoustic units is the task of finding the phonetic structure of a language from only speech data. No text data or any language-specific knowledge is assumed to be available. The task can be divided into three sub-tasks: segmentation, clustering segments, and modeling the sound pattern of each cluster [50]. In previous work, the three sub-problems were often approached sequentially and independently in which initial steps are not related to later ones [113, 50, 17]. For example, the speech data were usually segmented first, and clusters of acoustic units were learned based on the segments. In a sequential approach, the speech segmentation was never refined regardless of the clustering results, which prevented knowledge learned in one part of the problem from helping with learning the other parts of the problem.

In contrast to previous methods, we approach the task by modeling the three sub-problems as well as the unknown set of acoustic units as latent variables in one nonparametric Bayesian model. More specifically, we formulate a Dirichlet Process mixture model [3, 42, 120, 35] where each mixture is a Hidden Markov Model (DPHMM) that is used to model an acoustic unit and to generate observed segments of that unit. Our model seeks the set of sub-word units, segmentation, clustering, and HMMs that best represent the observed speech data through an iterative inference process, which is implemented by using Gibbs sampling. When tested on a corpus of English sentences, our model is able to discover sub-word units that are highly correlated with English phones, and also produces better segmentation than the state-of-the-art unsupervised baseline.

This DPHMM framework for unsupervised acoustic unit discovery is the foundation of this thesis. Building upon this model, we develop a computational model for discovering hierarchical linguistic structures from acoustic signals, and extend our investigation in two directions: *language acquisition* and *unsupervised ASR training*. In Section 1.3, we discuss the problem of acquiring phones and words that infants must solve when learning their first language, and briefly describe our model for joint discovery of phonetic and lexical units from continuous speech. In Section 1.4, we give an overview on the problem of one-shot learning and discuss the role of the phonetic compositional structure of a language in the task of learning new spoken words from just one or a few examples. In Section 1.5, we turn our focus to ASR and discuss the challenges of building speech recognizers for new languages that are posed by the current training procedure, and provide solutions to overcome the challenges.

■ 1.3 Hierarchical Linguistic Structure Discovery from Speech

Humans exchange knowledge, share experience, express opinions, and convey ideas via speech everyday. This verbal use of language is a hallmark of intelligence and is arguably the most unique innate ability of human beings [148, 179]. Language acquisition is an extremely complex process that involves an enormous amount of perceptual, computational, social and neural activity [97], which makes modeling this process with machines a truly daunting task. However, although most of the mechanism behind language acquisition remains unknown, research in cognitive science has shown that for some parts of the process, infants seem to rely on statistical clues for learning, and this is where probabilistic modeling can come into play. In this thesis, we focus our discussion on one particular problem of this type: *phonetic and lexical unit learning*, in which computational learning strategies are involved.

As pointed out in [96, 124, 88, 160, 159, 64], success in some sub-problems of the language acquisition procedure relies on infants' sensitivity to the *distributional patterns of sounds* in a language. For example, 9-month old infants can learn sequential constraints on permissible strings of phonemes in ambient language and discriminate between frequent phonetic sequences and less frequent ones [88]. Likewise, the transitional probabilities between adjacent syllables, which differ within and across words, can also help infants detect word boundaries in continuous speech [160, 159]. These statistical signals are important clues that help infants acquire phonetic categories and word types.

In addition to being an important source of information for infants, distributional patterns are also important learning constraints for probabilistic models. Much prior research has successfully modeled various parts of the language acquisition process by leveraging distributional patterns observed in natural language [60, 13, 25, 174, 33, 34, 82]. For example, by encoding word co-occurrence patterns and phonotactic constraints as grammar rules, the authors of [82] were able to apply *adaptor grammars* to effectively discover syllable and word boundaries from unsegmented phonetic transcripts of child-directed speech [83]. In fact, adaptor grammars have proven to be a powerful probabilistic framework for word segmentation [80, 82, 77], given their flexibility for modeling hierarchical linguistic structures, and their effectiveness for learning units of generalization from training data.

To learn hierarchical linguistic structure from speech data, we employ adaptor grammars as the basis of our approach. However, despite their effectiveness, adaptor grammars have only been developed for learning structures from symbolic input. In this thesis, we go beyond the original design of adaptor grammars and construct a novel probabilistic framework, which

not only discovers acoustic units, but also learns higher level linguistic structures such as syllabic and lexical units from continuous speech. Learning on this framework is supported by a Metropolis-Hastings-based inference procedure, which empowers synergies in unsupervised acquisition of phonetic and lexical units from speech. When tested on lecture recordings of various topics [59], our model demonstrates its ability to discover informative spoken keywords that occur frequently for each lecture.

■ 1.4 One-shot Learning of Spoken Words

The ability to learn new spoken words from only a few examples is an essential ingredient for language development. Most previous related computational work has focused on the problem of learning the meaning of words from a few examples. For instance, in [186], children were tested to decide which objects belong to the set of *elephants* and which do not after hearing the word *elephant* paired with an exemplar. Various factors such as cross-situational learning that may contribute to learning word meaning were also investigated in previous work [170, 47]. However, by any account, the acquisition of meaning is possible only if a child can learn the *spoken word as a category*, mapping all instances (and excluding non-instances) of a word like *elephant* to the same phonological representation, and this is the focus of the discussion on one-shot learning presented in this thesis. Particularly, we investigate the representation that humans may use to accomplish one-shot learning of spoken words.

Although only acquisition of new spoken words is discussed in this thesis, in reality, humans learn new concepts from just one or a few examples, making meaningful generalizations that go far beyond the observed examples in all kinds of scenarios. Replicating this ability in machines is challenging. Nonetheless, recent advances in cognitive science and machine learning have brought new insights into the mechanism behind one-shot learning. Particularly, the idea of developing new concepts from previous learning experience with related examples is a prominent theme. This idea of *learning-to-learn* can be realized as probabilistic models that first induce intrinsic structures in the data of interests, and then generalize for new data based on the induced intrinsic structures. For example, in [104], a hierarchical Bayesian model is exploited to learn the intrinsic compositional structure embedded in handwritten characters of fifty languages. The model infers primitive strokes that are shared across all the characters from a large number of handwritten examples, as shown in Fig. 1.2-(a)-i. These intrinsic structures embedded in all the characters can then be combined to create characters that are either familiar or new to the model, as shown in (ii) and (iii) of Fig. 1.2-(a), which thus allow the model to

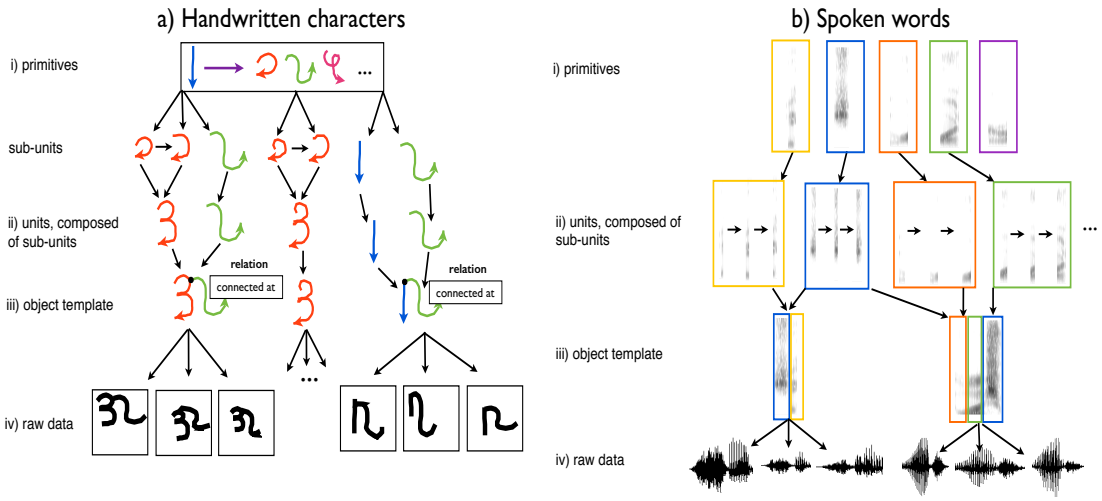


Figure 1.2. Hierarchical Bayesian modeling as applied to (a) handwritten characters [104] and (b) speech. Color coding highlights the re-use of primitive structure across different objects. The speech primitives are shown as spectrograms.

generalize to new concepts. As presented in [104], the model is able to achieve human-level performance for classifying new handwritten characters by learning from only one example.

The key to the effectiveness of the hierarchical Bayesian model presented in Fig. 1.2-(a) for one-shot learning of handwritten characters is that the model first learns the compositional structure shared across all different type of characters and acquires a library of basic strokes of written characters. These basic strokes can be regarded as the knowledge that the model has learned from its previous exposure to character data, which can then help the model learn and generalize to novel characters. We adopt the same learning-to-learn idea and investigate the importance of acquiring compositional structures for the task of one-shot learning of spoken words. More specifically, we apply our framework for unsupervised acoustic unit discovery to infer the basic acoustic units in a set of speech data, as illustrated in (i) and (ii) of Fig. 1.2-(b). These acoustic units can be viewed as knowledge the model has learned about the composition within spoken words as depicted in Fig. 1.2. The model can interpret new spoken words as new compositions of these basic acoustic units and thus generalize based on what it has learned from other speech data. We evaluate the model on both one-shot classification and one-shot generation of new spoken Japanese words and compare the performance our model to that of humans. The experimental results show that learning compositional structures in speech helps one-shot learning of spoken words even when the new words are spoken in a new language.

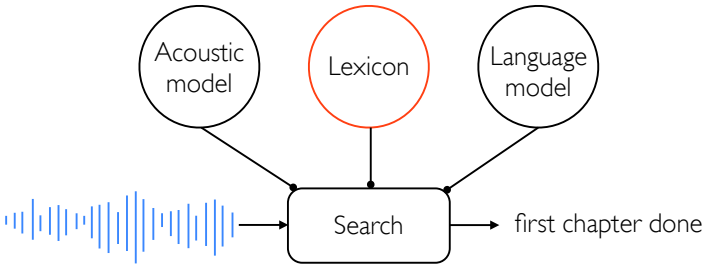


Figure 1.3. An overview of a typical ASR system, which contains an acoustic model, a pronunciation lexicon, and a language model. When the recognizer receives a speech input, it searches for the best word hypothesis based on the three components. The focus of a part of the thesis is on the lexicon, which consists of a list of word pronunciations of a language. The creation of a pronunciation lexicon remains the most inefficient process in developing a speech recognizer. Therefore, in this thesis, we present a framework that automatically discovers word pronunciations from parallel text and speech data.

This demonstrates an example of learning-to-learn through transferring knowledge of phonetic structure across languages.

■ 1.5 Pronunciation Learning for Unsupervised ASR Training

The effortless first language acquisition process demonstrated by humans stands in stark contrast to the highly supervised approach for training ASR systems. The basic ASR training procedure requires a large corpus of annotated speech data that includes audio waveforms and the corresponding orthographic transcriptions. In addition, we need a pronunciation lexicon that consists of a list of words and their associated phonemic pronunciations. With the required data, three major components of a speech recognizer can then be trained: (1) acoustic models can be built to model the speech realizations of phonetic units using the lexicon and the annotated speech data, (2) pronunciation models can be constructed based on the lexicon entries, and finally (3) language models can be trained upon the transcriptions. An illustration of a typical ASR system is shown in Fig. 1.3. Though various training methods exist, the development of a speech recognizer generally follows this standard recipe.

The requirement of a corpus of annotated speech data and a pronunciation lexicon is a significant impediment to the creation of speech recognizers for new languages. A generous estimate of the current language capacity of ASR systems would be roughly 100 to 150 out of the nearly 7,000 languages that are spoken all around the world [151]. Fortunately, crowdsourcing platforms such as Amazon Mechanical Turk have recently made collection of annotated

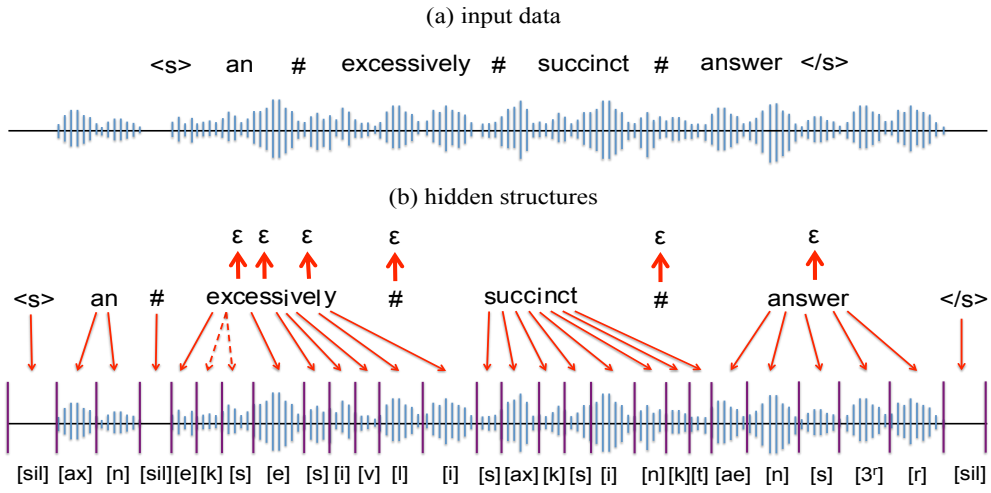


Figure 1.4. (a) An example of the input data and (b) the hidden structures embedded in the input data that need to be discovered by our model. The red arrows show the mapping between graphemes and phones. The thick arrows indicate that a grapheme is not mapped to any sounds, which is denoted as ϵ. The dotted arrows indicate that a grapheme is mapped to two sub-words at a time, and the solid arrows indicate that a grapheme is mapped to exactly one sub-word. The phone transcription is the sequence denoted by $[\cdot]$.

speech corpora a less expensive and more efficient task [126, 110, 107, 14, 123]. Although the traditional approach may still be preferred for certain low-resource languages [54], this new data collection paradigm has proven that annotated corpora can be relatively easily created by native speakers of a language. In contrast, as annotators must understand the subtle differences between distinct phonetic units, and familiarize themselves with the labeling system, the creation of a pronunciation lexicon demands a large amount of linguistic knowledge and cannot be easily crowdsourced. Given the nature of its production, the pronunciation lexicon is arguably the main hurdle in developing an ASR system. If word pronunciations can be automatically inferred from an annotated corpus, then the costly and time-consuming process of manually creating a lexicon can be avoided, and ASR systems can potentially be more efficiently deployed for many more languages.

We investigate the problem of inferring a pronunciation lexicon from an annotated corpus without exploiting any language-specific knowledge. A typical training sample is shown in Fig. 1.4-(a), where we have access to only the speech data along with the corresponding word level transcription. Our goal is to discover the basic phonetic structure hidden in the speech data, indicated by the vertical purple bars and the phone transcriptions in Fig. 1.4-(b),

as well as the latent encoding scheme between the graphemes of the language and the discovered phonetic units, which is indicated by the red arrows in Fig. 1.4-(b). Note that the phone transcription shown in Fig. 1.4-(b) is only for illustration as our model does not have access to the true phonetic labels during training. We formulate our approach as a hierarchical Bayesian model, which jointly discovers these two latent structures. Having obtained these two pieces of information, we can then convert sequences of graphemes in the language into sequences of phonetic units and thus create a word pronunciation lexicon for the language. We evaluate the quality of the induced lexicon and acoustic units through a series of speech recognition experiments on a conversational weather query corpus [195]. The results demonstrate that our model consistently performs closely to recognizers that are trained with an expert-defined phonetic inventory and lexicon.

■ 1.6 Thesis Contributions

The primary contributions of this thesis are threefold.

1. **Unsupervised linguistic structure discovery from acoustic signals:** We develop a set of unsupervised models for discovering linguistic structures directly from acoustic signals. In particular, we are the first to apply Dirichlet process mixture models to the task of acoustic unit discovery. Furthermore, we construct a novel modification to adaptor grammars such that the adaptor grammars can infer linguistic structures directly from speech signals. Our research results demonstrate that the proposed unsupervised models are capable of discovering highly meaningful linguistic units, which can potentially be applied to tasks such as keyword spotting, spoken document summarization, and information retrieval.

2. **Representation for capturing one-shot learning behavior:** We verify that learning compositional structure embedded in acoustic signals is important for learning new spoken words, which resonates with the findings of previous work on other one-shot learning tasks. Furthermore, our model shows that phonetic knowledge can be transferred across languages for learning words in a new language. Even though replicating human one-shot learning capability still remains a challenging task, our work suggests the type of structure to learn for capturing how humans learn rich concepts from very sparse data.

3. **Automatic pronunciation lexicon creation:** We invent a framework for learning word pronunciations from parallel speech and text data without the need for any language-

specific knowledge. Our work provides a practical solution to the expensive and time-consuming process of creating pronunciation lexicons that are necessary for training phonetic ASR systems.

The remarkable ability possessed by humans for first language acquisition is one of the major inspirations of this work. However, given the complexity of the whole language acquisition process, we refrain from claiming any solution to the grand problem. Nonetheless, we believe the success our models demonstrate in discovering linguistic structures directly from sensory speech signals has unlocked the door to a broad avenue for future research on the language acquisition process.

■ 1.7 Chapter Overview

The remainder of this thesis is organized as follows:

- Chapter 2 discusses the problem of unsupervised acoustic unit discovery and presents our approach based on the Dirichlet process mixture model for the task.
- Chapter 3 presents a model, based on an adaptation of adaptor grammars and the acoustic unit discovery framework, for learning hierarchical linguistic structures, including phonetic, syllabic, and lexical units, from speech data.
- Chapter 4 describes a variation of the acoustic unit discovery model and shows how to apply the modified model to investigate the role of the compositional structure in speech for one-shot learning of spoken words.
- Chapter 5 shows our approach to the joint learning of acoustic units and the grapheme-to-phone encoding scheme of a language from annotated speech data. We also demonstrate how to apply the learning results of the model to automatically produce a word pronunciation lexicon, which can then be used to build phonetic ASR systems.
- Chapter 6 summarizes the key points of this thesis and suggests possible directions for future work.

Chapter 2

Acoustic Unit Discovery in Speech

2.1 Chapter Overview

Unsupervised discovery of meaningful sub-word units from the speech signal has attracted much interest in the fields of Automatic Speech Recognition (ASR) and cognitive science. For ASR, the problem is interesting because it has the potential to change the current highly-supervised approach for building speech recognizers. Take the acoustic model as an example. The standard process of training an acoustic model for a language requires not only language-specific knowledge such as the phone set of the language, but also a large amount of transcribed speech data. Unfortunately, these necessary data are only available for a very small number of languages in the world. Therefore, if meaningful acoustic units can be automatically discovered from speech data, then acoustic models for a much larger number of languages in the world can be efficiently trained. In addition, these acoustic models alone can be applied to many problems such as Spoken Term Detection (STD), spoken document retrieval, and language identification [111, 44, 86, 105, 53]. Furthermore, as demonstrated in Chapter 5, with some word transcriptions for speech data, we can learn a pronunciation lexicon for a language based on automatically acquired acoustic units, which enables a fully automated method for training phone-based speech recognizers.

Unsupervised unit discovery is also interesting for the field of cognitive science because learning meaningful acoustic units from speech is one of the first challenges that an infant must face when he or she acquires his or her native language. There has been much research studying how infants acquire their first language; however, most of the works are based on highly processed input data such as phone transcriptions of child-directed speech or isolated speech segments of phones [119, 60, 40, 41, 24, 181, 33, 34]. A framework for unsupervised acoustic unit discovery enables us to more closely study how infants acquire a language from speech. In addition, as shown in Chapter 4, learning the basic acoustic units embedded in

speech is important for one-shot learning of spoken words, which is arguably the most essential step for humans to learn new concepts about the world.

In this chapter, we investigate the problem of unsupervised acoustic unit discovery using only spoken utterances as training data. In other words, neither prior language-specific knowledge, such as the phone set of a language, nor transcribed data are available for training. We present an unsupervised model that simultaneously segments the speech, discovers a proper set of sub-word units, and learns a Hidden Markov Model (HMM) for each induced acoustic unit. Our approach is formulated as a Dirichlet process mixture model in which each mixture is an HMM that represents a sub-word unit. Our model seeks the set of sub-word units, segmentation, clustering, and HMMs that best represent the observed speech data through an iterative inference process based on Gibbs sampling.

We test our model on the TIMIT corpus [52], and the results demonstrate that our model discovers sub-word units that are highly correlated with standard English phones and also produces better segmentation than the state-of-the-art unsupervised baseline for the task of speech segmentation. We evaluate the quality of the learned acoustic units on an STD task. Compared to the baselines, our model improves the relative precision of top hits by at least 22.1%, and outperforms a language-mismatched acoustic model that is trained with the conventional highly-supervised method.

The rest of this chapter is organized as follows. In Section 2.2, we review three lines of research that are related to unsupervised acoustic unit discovery from speech data. We formalize the problem and introduce the observed and latent variables for the problem in Section 2.3. We present our model and describe the generative process implied by our model in Section 2.4. The inference algorithm based on Gibbs sampling for learning our model is shown in Section 2.5. We evaluate the model both qualitatively and quantitatively, and we describe the experimental setup in Section 2.6 as well as demonstrate the results in Section 2.7. Lastly, this chapter concludes in Section 2.8.

■ 2.2 Related Work

There are three lines of research that are related to unsupervised acoustic unit discovery: 1) unsupervised sub-word modeling, 2) unsupervised speech segmentation and 3) nonparametric Bayesian methods for segmentation and clustering. We review related work in each of the three categories and briefly compare our work to the previous approaches.

2.2.1 Unsupervised Sub-word Modeling

As suggested in [50], unsupervised acoustic modeling can be broken down into three sub-tasks: segmentation, clustering segments, and modeling the sound pattern of each cluster. We follow this general guideline, which is also used in [113, 50, 17], and approach the problem of unsupervised acoustic modeling by solving three sub-problems of the task. Even though our work, as well as the previous work of [113, 50, 17], all adopt the same guidelines, the key difference is that our model does not assume independence among the three aspects of the problem. Particularly, in previous work, the three sub-problems were often approached sequentially and independently in which initial steps are not related to later ones. For example, the speech data are usually segmented regardless of the clustering results and the learned acoustic models. On the contrary, we approach the problem by modeling the three sub-problems as well as the unknown set of sub-word units as latent variables in one nonparametric Bayesian model. Therefore, our model can refine its solution to one sub-problem by exploiting what it has learned about other parts of the problem. Second, unlike [113, 50] in which the number of sub-word units to be learned is assumed to be known, our model learns the size from the training data directly.

Instead of segmenting utterances, the authors of [182] train a single state HMM using all data at first, and then iteratively split the HMM states based on an objective function. This method achieves high performance in a phone recognition task using a label-to-phone transducer trained from some transcriptions. However, the performance seems to rely on the quality of the transducer. For our work, we assume no transcriptions are available and measure the quality of the learned acoustic units via a spoken query detection task as in [74]. The authors of [74] approach the task of unsupervised acoustic modeling by first discovering repetitive patterns in the data, and then learn a whole-word HMM for each found pattern, where the state number of each HMM depends on the average length of the pattern. The states of the whole-word HMMs are then collapsed and used to represent acoustic units. This approach implicitly imposes information from the word-level on learning sub-word units, which provides a stronger learning constraint than what our model has. More specifically, instead of discovering repetitive patterns first as in [74], our model learns from all given speech data.

Although there has been computational work on the problem of unsupervised learning of sub-word units in the field of cognitive science [181, 40, 41, 24], most of these models cannot be directly applied to speech input. These models usually assume that the phonetic boundaries in speech are known, and that the speech data are already converted in a low-dimensional space such as the first and second formant of vowel sounds. In contrast, our model infers sub-word segmentation and sub-word categories from a feature representation that more closely

resembles raw speech data.

2.2.2 Unsupervised Speech Segmentation

As mentioned in Section 2.2.1, one goal of our model is to segment speech data into small sub-word (e.g., phone) segments. Since segmentation performance is one of the metrics we use to evaluate the model, we review previous unsupervised speech segmentation methods in this section. In general, most unsupervised speech segmentation methods rely on acoustic change for hypothesizing phone boundaries [165, 152, 28, 37, 56]. Even though the overall approaches differ, these algorithms are all one-stage and bottom-up segmentation methods [165]. In contrast, given that segmentation is only one of the three tasks that our model jointly tackles, it does not make a single one-stage decision. Instead, the model infers the segmentation through an iterative process, and exploits the learned sub-word models to guide its hypotheses on phone boundaries.

2.2.3 Nonparametric Bayesian Methods for Segmentation and Clustering

Our model is inspired by previous applications of nonparametric Bayesian models to segmentation and clustering problems in natural language processing and speaker diarization [60, 45]; particularly, we adapt the inference method used in [60] to train our model. The unsupervised acoustic unit discovery problem is, in principle, similar to the word segmentation problem discussed in [60]. In the word segmentation problem, sequences of phone transcriptions of spoken utterances are given, and the goal of the model presented in [60] is to find the word boundaries within each phone sequence. As for our problem, sequences of speech features are given, and the goal of our model is to find the phone boundaries within each feature sequence. The main difference, however, is that our model is under the continuous real value domain, and the problem of [60] is under the discrete symbolic domain. Therefore, the model of [60] does not need to cluster the induced word segments since the cluster label of each segment is simply the sequence of phones it carries. For the domain our problem is applied to, our model has to include the cluster label of each phone segment as a latent variable, and thus the learning for our model is more complex.

2.3 Problem Formulation

The goal of our model, given a set of speech utterances, is to jointly learn the following:

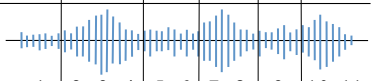
Pronunciation	b [b]	a [ax]	n [n]	a [ae]	n [n]	a [ax]
						
Frame index (t)	1	2 3 4	5 6	7 8	9	10 11
Speech feature (x_t^i)	x_1^i	$x_2^i x_3^i x_4^i$	$x_5^i x_6^i$	$x_7^i x_8^i$	x_9^i	$x_{10}^i x_{11}^i$
Boundary variable (b_t^i)	1	0 0 1	0 1	0 1	1	0 1
Boundary index (g_q^i)	g_0^i g_1^i	g_2^i	g_3^i	g_4^i	g_5^i	g_6^i
Segment $(p_{j,k}^i)$	$p_{1,1}^i$	$p_{2,4}^i$	$p_{5,6}^i$	$p_{7,8}^i$	$p_{9,9}^i$	$p_{10,11}^i$
Duration $(d_{j,k}^i)$	1	3	2	2	1	2
Cluster label $(c_{j,k}^i)$	$c_{1,1}^i$	$c_{2,4}^i$	$c_{5,6}^i$	$c_{7,8}^i$	$c_{9,9}^i$	$c_{10,11}^i$
HMM (θ_c)	θ_1	θ_2	θ_3	θ_4	θ_3	θ_2
Hidden state (s_t^i)	1	1 2 3	1 3	1 3	1	1 3
Mixture ID	1	1 6 8	3 7	5 2	8	2 8

Figure 2.1. An example of the observed data and hidden variables of the problem for the word *banana*. See Section 2.3 for a detailed explanation.

1. Segmentation: To find the phonetic boundaries within each utterance (i.e., to find segments).

2. Nonparametric clustering: To find a proper set of clusters and group acoustically similar segments into the same cluster (i.e., to find sub-word units).

3. Sub-word modeling: To learn an HMM to model each sub-word acoustic unit.

We model the three sub-tasks as latent variables in our approach. In this section, we describe the observed data, latent variables, and auxiliary variables of the problem and show an example in Fig. 2.1. In the next section, we show the generative process our model uses to generate the observed data.

■ 2.3.1 Observed and Latent Variables

- **Speech Feature** (x_t^i)**:** The only observed data for our problem are a set of spoken utterances, which are converted to a series of 25 ms 13-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) [23], and their first- and second-order time derivatives at a 10 ms analysis rate. We use $x_t^i \in \mathbb{R}^{39}$ to denote the t^{th} feature frame of the i^{th} utterance.

Fig. 2.1 illustrates how the speech signal of a single word utterance *banana* is converted to a sequence of feature vectors x_1^i to x_{11}^i.

- **Boundary (b_t^i):** We use a binary variable b_t^i to indicate whether a phone boundary exists between x_t^i and x_{t+1}^i. If our model hypothesizes x_t^i to be the last frame of a sub-word unit, which is called a *boundary frame*, b_t^i is assigned with value 1, or it is assigned 0 otherwise. Fig. 2.1 shows an example of the boundary variables where the values correspond to the true answers. We use an auxiliary variable g_q^i to denote the index of the q^{th} boundary frame in utterance i. For example, in Fig. 2.1, $g_2^i = 4$. To make the derivation of posterior distributions easier in Section 2.5, we define g_0^i to be the beginning of an utterance, and L_i to be the number of boundary frames in an utterance. For the example shown in Fig. 2.1, L_i is equal to 6.

- **Segment ($p_{j,k}^i$):** We define a segment to be composed of feature vectors between two boundary frames. We use $p_{j,k}^i$ to denote a segment that consists of $x_j^i, x_{j+1}^i \cdots x_k^i$ and $d_{j,k}^i$ to denote the length of $p_{j,k}^i$. See Fig. 2.1 for more examples.

- **Cluster Label ($c_{j,k}^i$):** We use $c_{j,k}^i$ to specify the cluster label of $p_{j,k}^i$. We assume segment $p_{j,k}^i$ is generated by the sub-word HMM with label $c_{j,k}^i$.

- **HMM (θ_c):** In our model, each HMM has three emission states, which correspond to the beginning, middle, and end of a sub-word unit [76]. A traversal of each HMM must start from the first state, and only left-to-right transitions are allowed even though we allow skipping of the middle and the last state for segments shorter than three frames. The emission probability of each state is modeled by a diagonal Gaussian Mixture Model (GMM) with 8 mixtures. We use θ_c to represent the set of parameters that define the c^{th} HMM, which includes state transition probability $a_c^{j,k}$, and the GMM parameters of each state emission probability. We use $w_{c,s}^m \in \mathbb{R}$, $\mu_{c,s}^m \in \mathbb{R}^{39}$, and $\lambda_{c,s}^m \in \mathbb{R}^{39}$ to denote the weight, mean vector, and diagonal of the inverse covariance matrix of the m^{th} mixture in the GMM for the s^{th} state in the c^{th} HMM.

- **Hidden State (s_t^i):** Since we assume the observed data are generated by HMMs, each feature vector, x_t^i, has an associated hidden state index. We denote the hidden state of x_t^i as s_t^i.

- **Mixture ID (m_t^i):** Similarly, each feature vector is assumed to be emitted by the state GMM it belongs to. We use m_t^i to identify the Gaussian mixture that generates x_t^i.

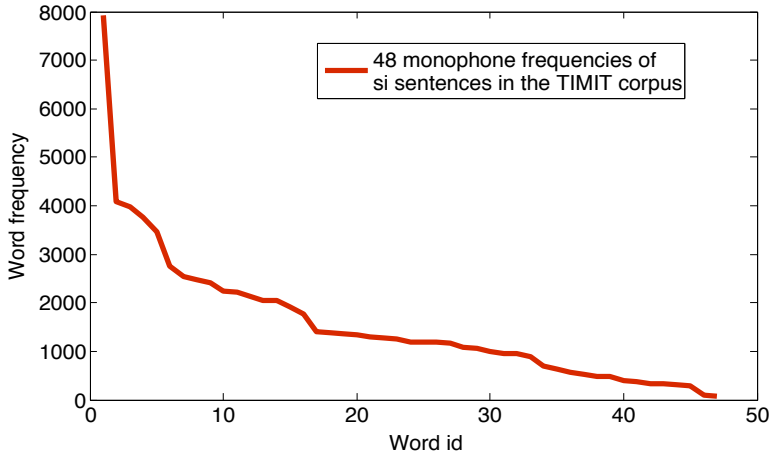


Figure 2.2. Monophone frequency distribution of the *si* sentences of the TIMIT corpus [52]. The most frequent monophone is the closure, denoted as /cl/ in the TIMIT corpus, that appears before the release of unvoiced stop consonants /t/, /p/ and /k/, and the least frequent monophone is /zh/.

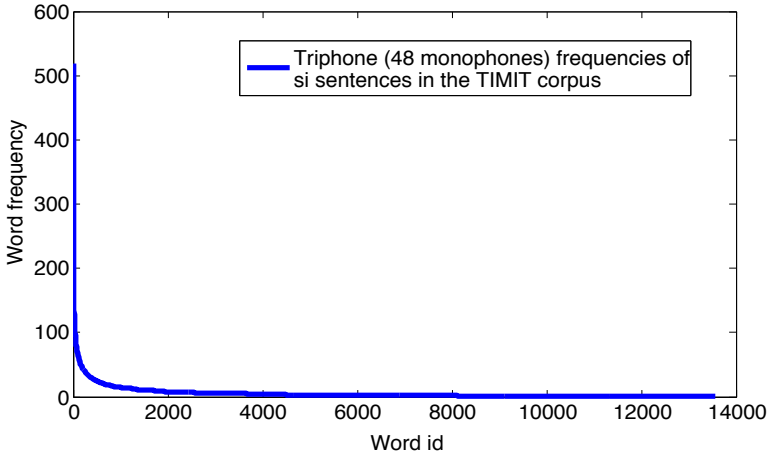


Figure 2.3. Triphone frequency distribution of the *si* sentences of the TIMIT corpus [52]. The most frequent triphone sequence is /s-cl-t/, which appears roughly 520 times in all the 1890 *si* sentences.

2.4 Model

2.4.1 Dirichlet Process Mixture Model with Hidden Markov Models

We aim to discover and model a set of sub-word units that represent the spoken data. If we think of utterances as sequences of repeated sub-word units, then in order to find the sub-words, we

need a model that concentrates probability on highly frequent patterns while still preserving some probability for previously unseen patterns. As a sanity check on the assumption that there are only a few frequently observed sub-word units, and many more rarely seen sub-words in real data, we plot the 48 monophone frequency distribution and the triphone frequency distribution of the *si* sentences in the TIMIT corpus in Fig. 2.2 and Fig. 2.3. As illustrated in the two figures, for both monophones and triphones, there are a few highly frequent units, while most of the units do not appear as often. Given that frequencies of the clusters induced by Dirichlet processes also exhibit the same *long tail* property, Dirichlet processes are particularly suitable for our goal[1]. Therefore, we construct our model as a Dirichlet Process (DP) mixture model, whose components are HMMs that are used to model sub-word units. We assume each spoken segment is generated by one of the clusters in this DP mixture model.

■ 2.4.2 Generative Process

Here, we describe the generative process our model uses to generate the observed utterances and present the corresponding graphical model. For clarity, we assume that the values of the boundary variables b_t^i are given in the generative process. In the next section, we explain how to infer their values.

Let $p^i_{g_q^i+1,g_{q+1}^i}$ for $0 \leq q \leq L_i - 1$ be the segments of the i^{th} utterance. Our model assumes each segment is generated as follows:

1. Choose a cluster label $c^i_{g_q^i+1,g_{q+1}^i}$ for $p^i_{g_q^i+1,g_{q+1}^i}$. This cluster label can be either an existing label or a new one. Note that the cluster label determines which HMM is used to generate the segment.

2. Given the cluster label, choose a hidden state for each feature vector x_t^i in the segment.

3. For each x_t^i, based on its hidden state, choose a mixture from the GMM of the chosen state.

4. Use the chosen Gaussian mixture to generate the observed feature vector x_t^i.

[1]Another widely used stochastic process in the field of natural language processing is the Pitman-yor process [150, 149], whose *discount parameter* empowers its flexibility on modeling the *long tail* distribution observed in many statistics embedded in human languages such as word frequencies. Since the Dirichlet process is a special case of the Pitman-yor process, it is straightforward to extend our model and the inference algorithm to use Pitman-yor process as a prior.

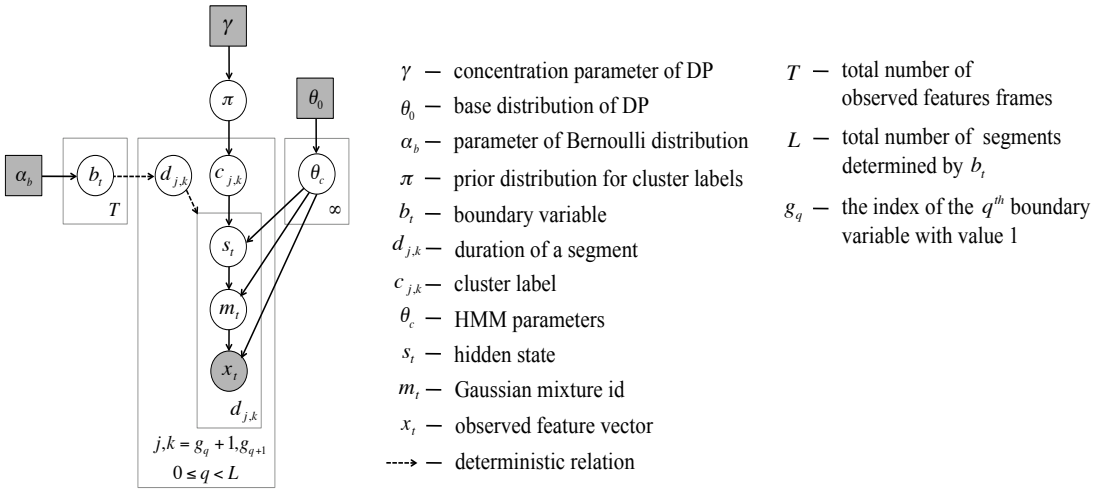


Figure 2.4. The graphical model for our approach. The shaded circle denotes the observed feature vectors, and the squares denote the hyperparameters of the priors used in our model. The dashed arrows indicate deterministic relations. Note that the Markov chain structure over the s_t variables is not shown here to keep the graph concise.

The generative process indicates that our model ignores utterance boundaries and views the entire data as concatenated spoken segments. Given this viewpoint, we discard the utterance index, i, of all variables in the rest of the chapter.

The graphical model representing this generative process is shown in Fig. 2.4, where the shaded circle denotes the observed feature vectors, and the squares denote the hyperparameters of the priors used in our model. Specifically, we use a Bernoulli distribution as the prior of the boundary variables and impose a Dirichlet process prior on the cluster labels and the HMM parameters. The dashed arrows represent deterministic relations. For example, the boundary variables deterministically construct the duration of each segment, d, which in turn sets the number of feature vectors that should be generated for a segment. In the next section, we show how to infer the value of each of the latent variables in Fig. 2.4 Note that the value of π in Fig. 2.4 is irrelevant to our problem; therefore, it is integrated out in the inference process.

■ 2.5 Inference

We employ Gibbs sampling [55] to approximate the posterior distribution of the hidden variables in our model. To apply Gibbs sampling to our problem, we need to derive the conditional posterior distribution of each hidden variable of the model. In the following sections, we first derive the sampling equations for each hidden variable, and then describe how we incorporate

acoustic cues to reduce the sampling load at the end.

■ 2.5.1 Sampling Equations for the Latent Variables

Here we present the sampling equations for each hidden variable defined in Section 2.3. We use $P(\cdot|\cdots)$ to denote a conditional posterior probability given observed data, all the other variables, and hyperparameters for the model.

Cluster Label ($c_{j,k}$) Let C be the set of distinct label values in $c_{-j,k}$, which represents all the cluster labels in training data except $c_{j,k}$. The conditional posterior probability of $c_{j,k}$ for $c \in C$ is:

$$\begin{aligned} P(c_{j,k} = c|\cdots) &\propto P(c_{j,k} = c|c_{-j,k};\gamma)P(p_{j,k}|\theta_c) \\ &= \frac{n^{(c)}}{N-1+\gamma}P(p_{j,k}|\theta_c) \end{aligned} \tag{2.1}$$

where γ is a parameter of the DP prior; the higher the value of γ, the more likely a segment will be associated with a new cluster. Intuitively, sub-word units in a language should be much sparser than the observed spoken segments; therefore, we set γ to a small number in our experiment as shown in Section 2.6.3. The first line of Eq. 2.1 follows Bayes' rule. The first term is the conditional prior, which is a result of the DP prior imposed on the cluster labels. The second term is the conditional likelihood, which reflects how likely the segment $p_{j,k}$ is generated by HMM$_c$. We use $n^{(c)}$ to represent the number of cluster labels in $c_{-j,k}$ taking the value c, and N to represent the total number of segments in the current segmentation.

In addition to existing cluster labels, $c_{j,k}$ can also take a new cluster label, which corresponds to a new sub-word unit. The corresponding conditional posterior probability is:

$$P(c_{j,k} \neq c, c \in C|\cdots) \propto \frac{\gamma}{N-1+\gamma}\int_\theta P(p_{j,k}|\theta)\,d\theta \tag{2.2}$$

To deal with the integral in Eq. 2.2, we follow the suggestions in [154, 136]. We sample an HMM from the prior (note that a new HMM is sampled for each segment), and compute the likelihood of the segment given the new HMM to approximate the integral. Finally, by normalizing Eq. 2.1 and Eq. 2.2, the Gibbs sampler can draw a new value for $c_{j,k}$ by sampling from the normalized distribution.

Hidden State (s_t) To enforce the assumption that a traversal of an HMM must start from the first state and end at the last state, we do not sample hidden state indices for the first and last

Algorithm 2.5.1 Initialization of s_t for the first inference iteration

for $j \leq t \leq k$ **do**
  if $t = j$ **then**
    $s_i^t = 1$
  else if $t = k$ **then**
    $s_i^k = 3$
  else
    $P(s_i^t = q | s_i^{t-1}) = a_{c_i^{j,k}}^{s_i^{t-1},q}$
  end if
end for

frames of a segment. If a segment has only one frame, we assign the first state to it. For each of the remaining feature vectors in a segment $p_{j,k}$, we sample a hidden state index according to the conditional posterior probability:

$$\begin{aligned} P(s_t = s | \cdots) &\propto P(s_t = s | s_{t-1}) P(x_t | \theta_{c_{j,k}}, s_t = s) P(s_{t+1} | s_t = s) \\ &= a_{c_{j,k}}^{s_{t-1},s} P(x_t | \theta_{c_{j,k}}, s_t = s) a_{c_{j,k}}^{s,s_{t+1}} \end{aligned} \tag{2.3}$$

where the first term and the third term are the conditional prior — the transition probability of the HMM that $p_{j,k}$ belongs to. The second term is the likelihood of x_t being emitted by state s of $\text{HMM}_{c_{j,k}}$. The variables s_{t-1} and s_{t+1} are the current state ids associated with x_{t-1} and x_{t+1}. Note that for initialization, s_t is sampled from the first prior term in Eq. 2.3. More specifically, the value of s_t in the first inference iteration is initialized by the process shown in Alg. 2.5.1.

Mixture ID (m_t) For each feature vector in a segment, given the cluster label $c_{j,k}$ and the hidden state index s_t, the derivation of the conditional posterior probability of its mixture ID is straightforward:

$$\begin{aligned} P(m_t = m | \cdots) &\propto P(m_t = m | \theta_{c_{j,k}}, s_t) P(x_t | \theta_{c_{j,k}}, s_t, m_t = m) \\ &= w_{c_{j,k},s_t}^m P(x_t | \mu_{c_{j,k},s_t}^m, \lambda_{c_{j,k},s_t}^m) \end{aligned} \tag{2.4}$$

where $1 \leq m \leq 8$. The conditional posterior consists of two terms: 1) the mixing weight of the m^{th} Gaussian in the state GMM indexed by $c_{j,k}$ and s_t and 2) the likelihood of x_t given the Gaussian mixture. The sampler draws a value for m_t from the normalized distribution of Eq. 2.4.

HMM Parameters (θ_c) Each θ_c consists of two sets of variables that define an HMM: the state emission probabilities $w_{c,s}^m, \mu_{c,s}^m, \lambda_{c,s}^m$ and the state transition probabilities $a_c^{j,k}$. In the following, we derive the conditional posteriors of these variables.

Mixture Weight $w_{c,s}^m$: We use $\underline{w}_{c,s} = \{w_{c,s}^m | 1 \leq m \leq 8\}$ to denote the mixing weights of the Gaussian mixtures of state s of HMM c. We choose a symmetric Dirichlet distribution with a positive hyperparameter β as its prior. The conditional posterior probability of $\underline{w}_{c,s}$ is:

$$\begin{aligned} P(\underline{w}_{c,s}|\cdots) &\propto P(\underline{w}_{c,s};\beta)P(\mathbf{m}_{c,s}|\underline{w}_{c,s}) \\ &\propto Dir(\underline{w}_{c,s};\beta)Mul(\mathbf{m}_{c,s};\underline{w}_{c,s}) \quad (2.5) \\ &\propto Dir(\underline{w}_{c,s};\beta') \quad (2.6) \end{aligned}$$

where $\mathbf{m}_{c,s}$ is the set of mixture IDs of feature vectors that belong to state s of HMM c. The m^{th} entry of β' is $\beta + \sum_{m_t \in \mathbf{m}_{c,s}} \delta(m_t, m)$, where we use $\delta(\cdot)$ to denote the discrete Kronecker delta. The last line of Eq. 2.6 comes from the fact that Dirichlet distributions are a conjugate prior for multinomial distributions. This property allows us to derive the update rule analytically.

Gaussian Mixture $\mu_{c,s}^m, \lambda_{c,s}^m$: We assume the dimensions in the feature space are independent. This assumption allows us to derive the conditional posterior probability for a single-dimensional Gaussian and generalize the results to other dimensions.

Let the d^{th} entry of $\mu_{c,s}^m$ and $\lambda_{c,s}^m$ be $\mu_{c,s}^{m,d}$ and $\lambda_{c,s}^{m,d}$. The conjugate prior we use for the two variables is a normal-Gamma distribution with hyperparameters μ_0, κ_0, α_0, and β_0 [134].

$$P(\mu_{c,s}^{m,d}, \lambda_{c,s}^{m,d}|\mu_0, \kappa_0, \alpha_0, \beta_0) = N(\mu_{c,s}^{m,d}|\mu_0, (\kappa_0\lambda_{c,s}^{m,d})^{-1})Ga(\lambda_{c,s}^{m,d}|\alpha_0, \beta_0)$$

By tracking the d^{th} dimension of feature vectors $x \in \{x_t | m_t = m, s_t = s, c_{j,k} = c, x_t \in p_{j,k}\}$, we can derive the conditional posterior distribution of $\mu_{c,s}^{m,d}$ and $\lambda_{c,s}^{m,d}$ analytically following the procedures shown in [134].

Transition Probabilities $a_c^{j,k}$: We represent the transition probabilities at state j in HMM c using $\underline{a}_c^j$. If we view $\underline{a}_c^j$ as mixing weights for states reachable from state j, we can simply apply the update rule derived for the mixing weights of Gaussian mixtures shown in Eq. 2.6 to $\underline{a}_c^j$. Assuming we use a symmetric Dirichlet distribution with a positive hyperparameter η as the prior, the conditional posterior for $\underline{a}_c^j$ is:

$$P(\underline{a}_c^j|\cdots) \propto Dir(\underline{a}_c^j;\eta')$$

where the k^{th} entry of η' is $\eta + n_c^{j,k}$, the number of occurrences of the state transition pair (j, k) in segments that belong to HMM c. Briefly, to compute the posterior distribution of a_c^j, we only need to track the number of times each transition pair (j, k) occurs among all segments that belong to cluster c and use these values to update the prior Dirichlet distribution.

Boundary Variable (b_t) To derive the conditional posterior probability for b_t, we introduce two variables:

$$l = (\arg\max_{g_q} g_q < t) + 1$$
$$r = \arg\min_{g_q} t < g_q$$

where l is the index of the closest turned-on boundary variable that precedes b_t plus 1, while r is the index of the closest turned-on boundary variable that follows b_t. Note that because g_0 and g_L are defined, l and r always exist for any b_t.

Our Gibbs sampler considers one boundary variable at a time while keeping the values of other boundary variables the same. Therefore, the value of b_t only affects the segmentation between x_l and x_r. If b_t is turned on, the sampler hypothesizes two segments $p_{l,t}$ and $p_{t+1,r}$ between x_l and x_r. Otherwise, only one segment $p_{l,r}$ is hypothesized. Since the segmentation on the rest of the data remains the same no matter what value b_t takes, the conditional posterior probability of b_t is:

$$P(b_t = 1|\cdots) \propto P(p_{l,t}, p_{t+1,r}|\mathbf{c}^-, \boldsymbol{\theta}) \tag{2.7}$$
$$P(b_t = 0|\cdots) \propto P(p_{l,r}|\mathbf{c}^-, \boldsymbol{\theta}) \tag{2.8}$$

where we assume that the prior probabilities for $b_t = 1$ and $b_t = 0$ are equal; $\mathbf{c}^-$ is the set of cluster labels of all segments except those between x_l and x_r ; and $\boldsymbol{\theta}$ indicates the set of HMMs that have associated segments. Our Gibbs sampler hypothesizes b_t's value by sampling from the normalized distribution of Eq. 2.7 and Eq. 2.8. The full derivations of Eq. 2.7 and Eq. 2.8 are shown in Eq. 2.9 and Eq. 2.10.

$$
\begin{aligned}
P(p_{l,t}, p_{t+1,r}|\mathbf{c}^-, \boldsymbol{\theta}) &= P(p_{l,t}|\mathbf{c}^-, \boldsymbol{\theta}) P(p_{t+1,r}|\mathbf{c}^-, c_{l,t}, \boldsymbol{\theta}) \\
&= \left[\sum_{c \in C} \frac{n^{(c)}}{N^- + \gamma} P(p_{l,t}|\theta_c) + \frac{\gamma}{N^- + \gamma} \int_\theta P(p_{l,t}|\theta)\, d\theta \right] \qquad (2.9) \\
&\times \left[\sum_{c \in C} \frac{n^{(c)} + \delta(c_{l,t}, c)}{N^- + 1 + \gamma} P(p_{t+1,r}|\theta_c) + \frac{\gamma}{N^- + 1 + \gamma} \int_\theta P(p_{t+1,r}|\theta)\, d\theta \right] \\
P(p_{l,r}|\mathbf{c}^-, \boldsymbol{\theta}) &= \sum_{c \in C} \frac{n^{(c)}}{N^- + \gamma} P(p_{l,r}|\theta_c) + \frac{\gamma}{N^- + \gamma} \int_\theta P(p_{l,r}|\theta)\, d\theta \qquad (2.10)
\end{aligned}
$$

Note that in Eq. 2.9 and Eq. 2.10, N^- is the total number of segments in the data except those between x_l and x_r. For $b_t = 1$, to account for the fact that when the model generates $p_{t+1,r}$, $p_{l,t}$ is already generated and owns a cluster label, we sample a cluster label for $p_{l,t}$ that is reflected in the Kronecker delta function. To handle the integral in Eq. 2.9 and Eq. 2.10, we sample one HMM from the prior and compute the likelihood using the new HMM to approximate the integral as suggested in [154, 136].

■ 2.5.2 Heuristic Boundary Elimination

To reduce the inference load on the boundary variables b_t, we exploit acoustic cues in the feature space to eliminate b_t's that are unlikely to be phonetic boundaries. We follow the pre-segmentation method described in [57] to achieve the goal. For the rest of the boundary variables that are proposed by the heuristic algorithm, we randomly initialize their values and proceed with the sampling process described above. Fig. 2.5 shows an example of applying the boundary elimination algorithm to a spoken utterance. It can be seen that only a small set of feature vectors, highlighted with vertical red bars in Fig. 2.5, are proposed as potential segment boundaries. Empirically, this boundary elimination algorithm can reduce the computational complexity of the inference algorithm described in this section by roughly an order of magnitude on clean speech corpora. In addition, as illustrated in Fig. 2.5, a subset of these proposed boundaries often coincide with true phone boundaries, which offers a good initialization for our model in the enormous hypothesis space.

■ 2.6 Experimental Setup

To the best of our knowledge, there are no standard corpora for evaluating unsupervised methods for acoustic modeling. However, numerous related studies have reported performance on

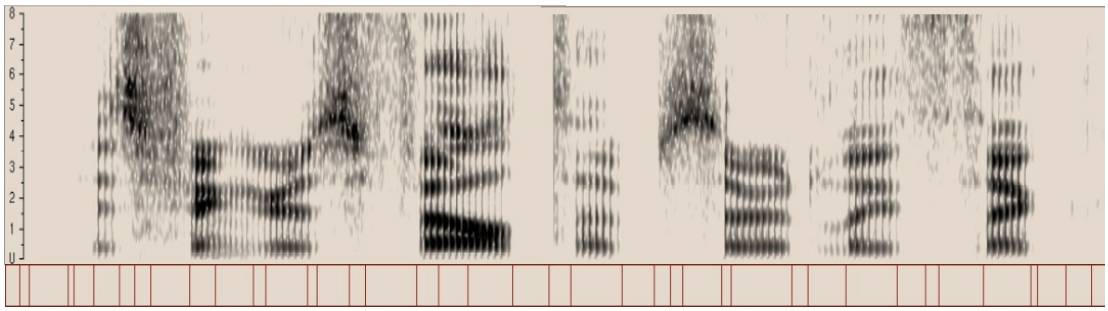


Figure 2.5. The result of applying the boundary elimination algorithm to a spoken utterance. The vertical red bars indicate potential segment boundaries proposed by the algorithm.

the TIMIT corpus [28, 37, 152, 191, 192], which creates a set of strong baselines for us to compare against. Therefore, the TIMIT corpus is chosen as the evaluation set for our model. In this section, we describe the TIMIT corpus and the methods used to measure the performance of our model on the following three tasks: sub-word acoustic modeling, segmentation, and nonparametric clustering.

■ 2.6.1 TIMIT Corpus

The TIMIT corpus is designed to provide speech data for acoustic-phonetic studies [52, 106, 194]. The corpus contains broadband recordings of 438 male speakers and 192 female speakers of eight major dialects of American English, each reading ten phonetically rich sentences. There are three sets of sentences in the TIMIT corpus: 1) the *sa* sentences that are read by every speaker, which are designed to reflect dialectal differences, 2) the *sx* phonetically compact sentences, which provide a good coverage of pairs of phones [106], and 3) the *si* phonetically diverse sentences, which contain sentences extracted from the Brown Corpus [95] and a collection of dialogs from stage plays [71]. The *si* sentences aim to add diversity in sentence types and phonetic contexts to the corpus.

The 6,300 sentences are further divided into training, dev, and test sets, which contain 4620, 500, and 1180 sentences. There is no speaker or sentence overlap between the training and test sets. The recorded utterances are stored as 16-bit, 16 kHz speech waveform files, and the time-aligned phonetic transcriptions of the recorded sentences are also provided in the corpus, which are the gold standard we use to evaluate the model.

■ 2.6.2 Evaluation Methods

Nonparametric Clustering

Our model automatically groups speech segments into different clusters. One question we are interested in answering is how these learned clusters correlate to English phones. To answer that question, we develop a method to map cluster labels to the phone set in a dataset. We align each cluster label in an utterance to the phone(s) it overlaps with in time by using the boundaries proposed by our model and the manually-labeled ones. When a cluster label overlaps with more than one phone, we align it to the phone with the largest overlap. An exception is when a cluster label is mapped to /vcl/ /b/, /vcl/ /g/, and /vcl/ /d/, where the duration of the release /b/, /g/, and /d/ is almost always shorter than the closure /vcl/, in which case we align the cluster label to both the closure and the release. We compile the alignment results for 3,696 training utterances, which consist of the TIMIT training set excluding all the *sa* subset, and present a confusion matrix between the learned cluster labels and the 48 phonetic units used in TIMIT [115].

Unsupervised Phone Segmentation

We compare the phonetic boundaries proposed by our model to the manual labels provided in the TIMIT dataset. We follow the suggestion of [165] and use a 20-ms tolerance window to compute recall, precision rates, and F-score of the segmentation our model proposed for TIMIT's training partition. We compare our model against the state-of-the-art unsupervised and semi-supervised segmentation methods that were also evaluated on the same set of data [28, 152].

Sub-word Modeling

Finally, and most importantly, we need to gauge the quality of the learned sub-word acoustic models. In previous work, [182] and [50] tested their models on a phone recognition task and a term detection task respectively. These two tasks are fair measuring methods, but performance on these tasks depends not only on the learned acoustic models, but also other components such as the label-to-phone transducer in [182] and the graphone model in [50]. To reduce performance dependencies on components other than the acoustic model, we turn to the task of spoken term detection, which is also the method used in [74].

We compare our unsupervised acoustic model with three supervised ones: 1) an English triphone model, 2) an English monophone model, and 3) a Thai monophone model. The first two were trained on TIMIT, while the Thai monophone model was trained from 32 hours of

Word	# occurences in the traning set	# occurences in the test set
age	3	8
warm	10	5
year	11	5
money	19	9
artists	7	6
problem	22	13
children	18	10
surface	3	8
development	9	8
organizations	7	6

Table 2.1. The ten keywords, used in the spoken term detection evaluation task, and their frequencies in the training and test sets of TIMIT.

clean, read Thai speech from the LOTUS corpus [90]. All of the three models, as well as ours, used three-state HMMs to model phonetic units. To conduct spoken term detection experiments on the TIMIT dataset, we computed a posteriorgram representation for both training and test feature frames over the HMM states for each of the four models. Ten keywords were randomly selected for the task. The list of the ten keywords and their counts of occurrences in the training and test sets are shown in Table 2.1. For every keyword, spoken examples were extracted from the training set and were searched for in the test set using segmental dynamic time warping [191].

In addition to the supervised acoustic models, we also compare our model against the state-of-the-art unsupervised methods for this task [191, 192]. [191] trained a GMM with 50 components to decode posteriorgrams for the feature frames, and [192] used a deep Boltzmann machine (DBM) trained with pseudo phone labels generated from an unsupervised GMM to produce a posteriorgram representation. The evaluation metrics they used were: 1) P@N, the average precision of the top N hits, where N is the number of occurrences of each keyword in the test set and 2) EER: the average equal error rate at which the false acceptance rate is equal to the false rejection rate. We also report experimental results using the P@N and EER metrics.

γ	α_b	β	η	μ_0	κ_0	α_0	β_0
1	0.5	3	3	$\boldsymbol{\mu}^d$	5	3	$3/\boldsymbol{\lambda}^d$

Table 2.2. The values of the hyperparameters of our model, where $\boldsymbol{\mu}^d$ and $\boldsymbol{\lambda}^d$ are the d^{th} entry of the mean and the diagonal of the inverse covariance matrix of training data.

■ 2.6.3 Hyperparameters and Training Details

The values of the hyperparameters introduced in Section 2.5 are shown in Table 2.2, where $\boldsymbol{\mu}^d$ and $\boldsymbol{\lambda}^d$ are the d^{th} entry of the mean and the diagonal of the inverse covariance matrix computed from training data. We pick these values to impose weak priors on our model. We run our sampler for 20,000 iterations, after which the evaluation metrics for our model all converged. In Section 2.7, we report the performance of our model using the sample from the last iteration.

■ 2.7 Results and Analysis

■ 2.7.1 Nonparametric Clustering

Fig. 2.6 shows a confusion matrix of the 48 phones used in TIMIT and the sub-word units learned from the 3,696 utterances of the TIMIT training set excluding the *sa* sentences. Each circle represents a mapping pair for a cluster label and an English phone. The confusion matrix demonstrates a strong correlation between the cluster labels and individual English phones. For example, clusters 19, 20, and 21 are mapped to the vowel /ae/. A more careful examination on the alignment results shows that the three clusters are mapped to the same vowel in a different acoustic context. For example, cluster 19 is mapped to /ae/ followed by stop consonants, while cluster 20 corresponds to /ae/ followed by nasal consonants. This context-dependent relationship is also observed in other English phones and their corresponding sets of clusters. Fig. 2.6 also shows that a cluster may be mapped to multiple English phones. For instance, cluster 89 is mapped to more than one phone; nevertheless, a closer look reveals that the cluster is mapped to /n/ and /d/, which are sounds with a similar place of articulation (i.e. dental). These correlations indicate that our model is able to discover the phonetic composition of a set of speech data without any language-specific knowledge.

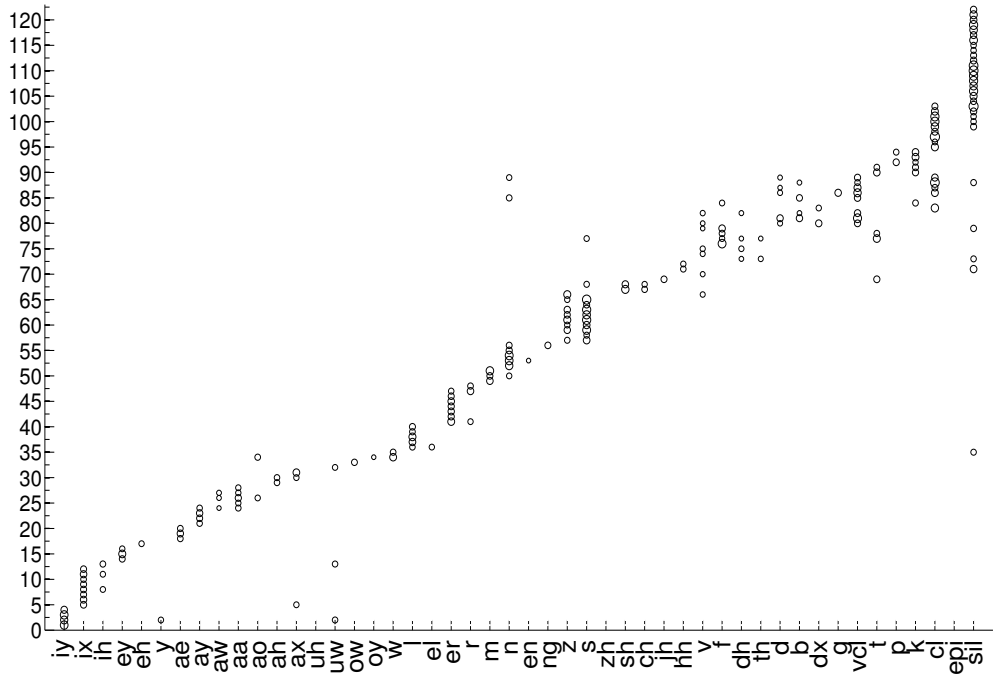


Figure 2.6. A confusion matrix of the learned cluster labels from the TIMIT training set excluding the sa type utterances and the 48 phones used in TIMIT. Note that for clarity, we show only pairs that occurred more than 200 times in the alignment results. The average co-occurrence frequency of the mapping pairs in this figure is 431.

■ 2.7.2 Sub-word Modeling

The performance of the four acoustic models on the spoken term detection task is presented in Table 2.3. The English triphone model achieves the best P@N and EER results and performs slightly better than the English monophone model, which indicates a correlation between the quality of an acoustic model and its performance on the spoken term detection task. Although our unsupervised model does not perform as well as the supervised English acoustic models, it generates a comparable EER, and achieves better detection performance for top hits than the Thai monophone model. This indicates that even without supervision, our model captures and learns the acoustic characteristics of a language automatically and is able to produce an acoustic model that outperforms a language-mismatched acoustic model trained with high supervision.

Table 2.4 shows that our model improves P@N by a large margin and generates only a slightly worse EER than the GMM baseline on the spoken term detection task. At the end of the training process, our model induced 169 HMMs, which were used to compute posteriorgrams.

unit(%)	P@N	EER
English triphone	75.9	11.7
English monophone	74.0	11.8
Thai monophone	56.6	14.9
Our model	63.0	16.9

Table 2.3. The performance of our model and three supervised acoustic models on the spoken term detection task.

unit(%)	P@N	EER
GMM [191]	52.5	16.4
DBM [192]	51.1	14.7
Our model	63.0	16.9

Table 2.4. The performance of our model, the GMM, and the state-of-the-art DBM baselines on the spoken term detection task for the TIMIT corpus.

This seems unfair at first glance because [191] only used 50 Gaussians for decoding, and the better result of our model could be a natural outcome of the higher complexity of our model. However, [191] pointed out that using more Gaussian mixtures for their model did not improve their model performance. This indicates that the key reason for the improvement is our joint modeling method instead of simply the higher complexity of our model.

Compared to the DBM baseline, our model produces a higher EER; however, it improves the relative detection precision of top hits by 24.3%. As indicated in [192], the hierarchical structure of DBM allows the model to provide a decent posterior representation of phonetic units. Even though our model only contains simple HMMs and Gaussians, it still achieves a comparable, if not better, performance than the DBM baseline. This demonstrates that even with just a simple model structure, the proposed learning algorithm is able to acquire rich phonetic knowledge from data and generate a fine posterior representation for phonetic units.

■ 2.7.3 Unsupervised Phone Segmentation

Table 2.5 summarizes the segmentation performance of the baselines, our model, and the heuristic pre-segmentation (pre-seg) method. The language-independent pre-seg method is suitable for seeding our model. It eliminates most unlikely boundaries while retaining about 87% true boundaries. Even though this indicates that at best our model only recalls 87% of the true boundaries, the pre-seg reduces the search space significantly. Additionally, the heuristic method is language-independent. Therefore, it can be integrated into our unsupervised learning

unit(%)	Recall	Precision	F-score
Dusan (2006) [unsupervised] [28]	75.2	66.8	70.8
Qiao (2008) [semi-supervised]* [152]	77.5	76.3	76.9
Our model [unsupervised]	76.2	76.4	76.3
Pre-seg	87.0	50.6	64.0

Table 2.5. The segmentation performance of the baselines, our model, and the heuristic pre-segmentation on TIMIT training set. *The number of phone boundaries in each utterance was assumed to be known in this model.

framework easily. Last but not least, it also allows the model to capture proper phone durations, which compensates for the fact that we do not include any explicit duration modeling mechanisms in our approach.

In the best semi-supervised baseline model [152], the number of phone boundaries in an utterance was assumed to be known. Although our model does not incorporate this information, it still achieves a very close F-score. When compared to the baseline in which the number of phone boundaries in each utterance was also unknown [28], our model outperforms it in both recall and precision, improving the relative F-score by 18.8%. The key difference between the two baselines and our method is that our model does not treat segmentation as a stand-alone problem; instead, it jointly learns segmentation, clustering, and acoustic units from data. The improvement on the segmentation task shown by our model further supports the strength of the joint learning scheme proposed in this chapter.

■ 2.8 Chapter Conclusion

In this chapter, we investigate and present a nonparametric Bayesian unsupervised approach to the problem of acoustic unit discovery from speech data. Without any prior knowledge, our method is able to discover phonetic units that are closely related to English phones, improve upon state-of-the-art unsupervised segmentation methods, and generate more precise spoken term detection performance on the TIMIT dataset. We see several directions to extend our work. First, can we use more flexible topological structures to model acoustic units within our framework as in [147]? For example, instead of fixing the number of states for each HMM to be three, can this number be directly learned from data? Second, while we assume no prior linguistic knowledge is available for training our model in the proposed learning framework, chances are that in reality we *will* have some knowledge about a language such as knowledge gained from a closely related language, or some universal phonological structure observed across well-

studied languages. How can we incorporate this information into the current model? Last but not least, as mentioned at the beginning of this chapter, discovering acoustic units from continuous speech is a task that infants must deal with when learning their first language. How can we compare the proposed model to humans? Can we make the proposed approach a cognitively plausible modeling framework for language acquisition?

The framework for unsupervised acoustic unit discovery presented in this chapter is the cornerstone of this thesis. In Chapter 3 to Chapter 5, we will use this framework as a foundation and venture out to investigate other problems that are related to automatic linguistic unit discovery from acoustic signals. Particularly, in Chapter 3, we present an unsupervised method, which is built on top of the framework depicted in this chapter, for acquiring phonetic, syllabic, and lexical structures from speech, the challenge that infants must solve when learning their mother tongue. In Chapter 4, we also apply this model to examine the role of the *compositional structure* in speech for the task of one-shot learning of spoken words. Finally, in Chapter 5, we demonstrate how to learn word pronunciations from a set of parallel text and speech data. The set of learned word pronunciations can then replace the manually-created lexicons used in modern ASR systems, which allows us to avoid the intensive human supervision involved in the current method for training speech recognizers.

Chapter 3

Hierarchical Linguistic Structure Discovery from Speech

3.1 Chapter Overview

In Chapter 2, we presented a DPHMM model for discovering phone-like acoustic units from speech. However, languages contain much richer structures that go beyond basic phonetic units. For instance, phone sequences form syllables, and syllable sequences form words. In this chapter, we investigate the problem of discovering hierarchical linguistic structures from speech data directly. In particular, we aim to learn not only the phonetic units but also the syllabic and the lexical units from acoustic signals. To accomplish the goal, we integrate an adaptor grammar with a noisy-channel model, which captures phonetic variability often observed in informal speech, and an acoustic model, which converts acoustic signals into symbolic phonetic units. This integrated framework is able to learn repeated, or reusable, acoustic patterns at all structural levels.

More concretely, when evaluated on lecture recordings, the model demonstrates an ability to discover lexical units that correspond to the set of words with high Term Frequency-Inverse Document Frequency (TFIDF) scores associated with each lecture. In addition, our model also demonstrates its capability of inducing phonetic units and discovering syllabic structures that can be reused to compose different lexical units, all directly from acoustic signals. An analysis on the experimental results reveals that modeling phonetic variability is the key to successfully acquiring lexical units from speech data. Last but not least, the analysis also shows that the synergies between phonetic and lexical unit learning helps improve the model performance.

The remainder of the chapter proceeds as follows. In Section 3.2, we discuss problems that are related to linguistic structure discovery from acoustic data and review the current approaches to tackling each of the related problems. In Section 3.3, we present our model and go

through the three main components of it: the adaptor grammars, the noisy-channel model, and the acoustic model in detail. After introducing the model, we show how to infer the latent variables by using Metropolis-Hastings algorithms in Section 3.4. In Section 3.5, we describe the dataset and the metrics used to evaluate the effectiveness of our model. The experimental results and analyses are reported in Section 3.6. Finally, we conclude the chapter in Section 3.7.

■ 3.2 Related Work

■ 3.2.1 Spoken Term Discovery

Our work is closely related to the Spoken Term Discovery (STD) problem, which is the task of discovering repeated lexical items from acoustic data. Previous work has approached this problem using pattern matching and graph clustering techniques [146, 191, 192, 75, 2, 127]. In particular, the authors of [146] modified the classic Dynamic Time Warping (DTW) algorithm to find acoustically similar speech segments within a dataset. To cluster the acoustic patterns, the segments were treated as adjacent nodes in a graph, connected by weighted edges indicating the similarity between the segments. Subsequently, graph clustering algorithms were employed to group the nodes into clusters based on acoustic similarity. The discovered clusters were shown to correspond to meaningful lexical entities such as words and short multi-word phrases in [146].

Building on this framework, the authors of [191, 192] proposed robust features that allowed lexical units to be discovered from spoken documents generated by different speakers. A similar approach to STD that was based on line segment detection on dotplots was also presented in [75]. Finally, instead of comparing every utterance pair to find recurrences of acoustic patterns as in [146, 191, 192, 75], an incremental comparison scheme was introduced in [127]. In [127], utterances were only compared to other utterances within a fixed recency window. Any utterance outside the window was represented only by the fragments already found within it. After the speech segments were all found incrementally, a graph clustering algorithm was then exploited to discover hidden categories of spoken words.

In [2], a multi-modal framework that utilized both speech and visual streams to discover lexical units in acoustic signals was proposed. The visual stream was abstracted as a sequence of discrete semantic tags, and each input spoken utterance was paired with one of these tags. When two utterances have the same visual semantic label, a Dynamic Programming (DP) algorithm is employed to search for acoustically similar segments within the pair. The discovered speech segments were then assigned to the cluster corresponding to the visual semantic tag. In

summary, instead of exploiting unsupervised clustering algorithms, the framework of [2] relied on the visual semantic label to cluster the speech segments.

Our model contrasts with the previous methods in two ways. First, due to the nature of the pattern matching techniques employed in the previous frameworks, no more than isolated speech segments that sparsely spread all over a dataset can be found by these methods. On the contrary, our model is able to induce continuous linguistic units embedded in the speech data. Furthermore, these methods can only discover shallow linguistic entities. For example, no structures within the discovered lexical acoustic units were learned by any of the methods. In contrast, our model is capable of acquiring hierarchical structures that contain rich linguistic information directly from acoustic signals.

■ 3.2.2 Word Segmentation on Symbolic Input

Much previous works has investigated the problem of segmenting continuous phoneme sequences into words. In particular, quite a few nonparametric Bayesian models have been proposed for the problem. In [60], the author applied both the Dirichlet Processes (DP) mixture model and the Hierarchical Dirichlet Processes (HDP) mixture model to discover latent word types in the input data. A nested Pitman-Yor language model was proposed to capture both the phoneme-ngram and the word-ngram statistics to find word boundaries in unsegmented phonetic input [131][1]. Even though the models proposed in [60] and [131] demonstrated their strength in finding word boundaries, these models all assumed an unrealistic degree of phonetic regularity. To overcome this drawback, a sequential and a joint model that are based on [60] were respectively proposed in [33, 34] for learning lexical units, and modeling phonetic variability from noisy phonetic input. Furthermore, the nested Pitman-Yor language model was also extended in [137, 69] to simultaneously learn lexical units and a language model from phoneme lattices representing speech signals.

Another model that is worth noting are adaptor grammars [83], which have proven to be a powerful framework for word segmentation. Particularly, adaptor grammars have been successfully applied to find word boundaries in various types of symbolic input, including phonetic transcriptions of speech signals, strings of Chinese characters, and unsegmented sequences of Sesotho characters [83, 80, 82, 77, 12, 78]. Briefly, adaptor grammars are a framework for defining a variety of hierarchical nonparametric Bayesian models. In fact, the DP mixture model proposed in [60] can be viewed as a special case of the adaptor grammars. The strength

[1]The Pitman-Yor language model was also used to learn the character-ngram and word-ngram to segment Japanese and Chinese character strings.

of adaptor grammars comes from their flexibility in learning units of generalization and the simplicity of encoding linguistic structures as grammars.

The major difference between our model and the methods discussed in this section is that our model infers linguistic units directly from acoustic signals. Particularly, unlike all the other methods, our model does not rely on any phonetic transcriptions that are generated either manually or by a pre-trained acoustic model. We achieve the goal by integrating adaptor grammars with a noisy-channel model and an acoustic model, which are described in more detail in Section 3.3.

■ 3.2.3 Linguistic Structure Discovery from Acoustic Signals

The goal of [20] is the most similar to ours: discovering linguistic structures directly from speech. The authors of [20] presented a two-level cascaded framework, in which one level of the model learns subword-like units, and the other level learns word-like units. Through an iterative optimization procedure, the model finds a set of acoustic patterns for each level and induces a language model for the word-like units from speech signals.

While our work and [20] share similar goals, there are two main differences that set our approach apart from theirs. First, in our approach, we explicitly model phonetic variability, which allows different phonetic realizations of a word to be mapped to the same lexical unit. Given that word-like units are defined as unique sequences of subword-like units in [20], it is not clear how their model clusters variant pronunciations of a word together. Second, by using adaptor grammars, we can easily encode linguistic structures that are richer than the two-level one defined in [20] and learn the corresponding composition from the speech data. For example, by modeling words as sequences of syllables, and representing syllables as sequences of phonetic units, we can infer acoustic patterns for three levels, which correspond to phonetic, syllabic, and lexical units. Last but not least, as mentioned earlier, adaptor grammars can learn units of generalization and encourage reuse of linguistic structures, which is a phenomenon frequently observed in natural language [143] and difficult to capture by a maximal likelihood learning framework such as the one proposed in [20].

■ 3.3 Model

■ 3.3.1 Problem Formulation and Model Overview

Given a corpus of spoken utterances, the goal of our model is to jointly infer the latent linguistic structures in each spoken utterance, which hierarchically consists of phonetic, syllabic, and

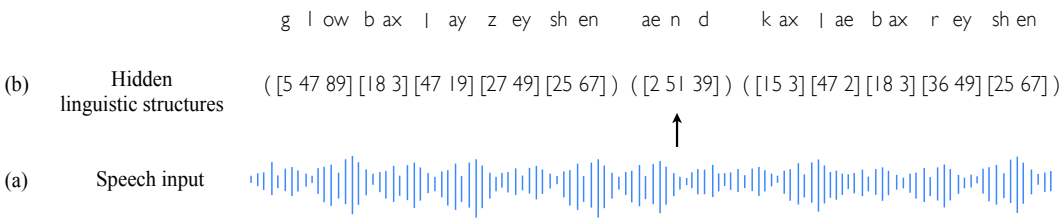


Figure 3.1. (a) Speech representation of the utterance *globalization and collaboration*, which is shown as a typical example of the input data to our model, and (b) the hidden linguistic structures that our model aims to discover that are embedded in the speech data, including the phonetic units (denoted as integers), syllabic units (indicated by $[\cdot]$), and lexical units (shown in $(\cdot)$). Note that the phone transcription, *g l ow b ax l ay z ey sh en ae n d k ax l ae b ax r ey sh en*, is only used to illustrate the structures our model aims to learn and is *not* given to our model for learning.

lexical units. Fig. 3.1-(a) and Fig. 3.1-(b) respectively show an input example and an illustration of the learning targets of our model using the utterance *globalization and collaboration*. The integers indicate the distinct phonetic units that our model learns from the speech data, and $[\cdot]$ and $(\cdot)$ denote the syllabic and lexical units. Note that the phone transcription, *g l ow b ax l ay z ey sh en ae n d k ax l ae b ax r ey sh en*, is only used to illustrate the structures our model aims to discover and is not available to our model for learning.

To discover hierarchical linguistic structures directly from acoustic signals, we divide the problem into three sub-tasks: 1) phonetic unit discovery, 2) phone variability modeling, and 3) syllabic and lexical unit learning. Each of the sub-tasks corresponds to some latent structures embedded in the speech data that our model needs to find. Here we briefly discuss the three sub-tasks as well as the latent variables associated with each sub-problem, and provide an overview on the proposed model for each of the sub-tasks.

Phonetic unit discovery For this sub-task, the goal of the model is to discover the phonetic units underlying each spoken utterance. In other words, the model aims to convert the speech input $\boldsymbol{x}_i$ into a sequence of Phone-Like Units (PLUs), $\vec{\boldsymbol{v}}_i$, which implicitly determines the phone segmentation, $\boldsymbol{z}_i$, in the speech data as indicated in (iv)-(vi) of Fig 3.2-(b). We use $\boldsymbol{x}_i = \{x_{i,t} | x_{i,t} \in \mathbb{R}^{39}, 1 \leq t \leq T_i\}$ to denote the series of Mel-Frequency Cepstral Coefficients (MFCCs) representing the i^{th} utterance [23], where T_i stands for the total number of feature frames in utterance i. In particular, we transform each spoken utterance into a series of 25 ms 13-dimensional MFCCs and their first- and second-order time derivatives at a 10 ms analysis rate. Each $x_{i,t}$ is associated with a binary variable $z_{i,t}$, indicating whether a PLU boundary exists between $x_{i,t}$ and $x_{i,t+1}$. The feature vectors with $z_{i,t} = 1$ are highlighted by the dark

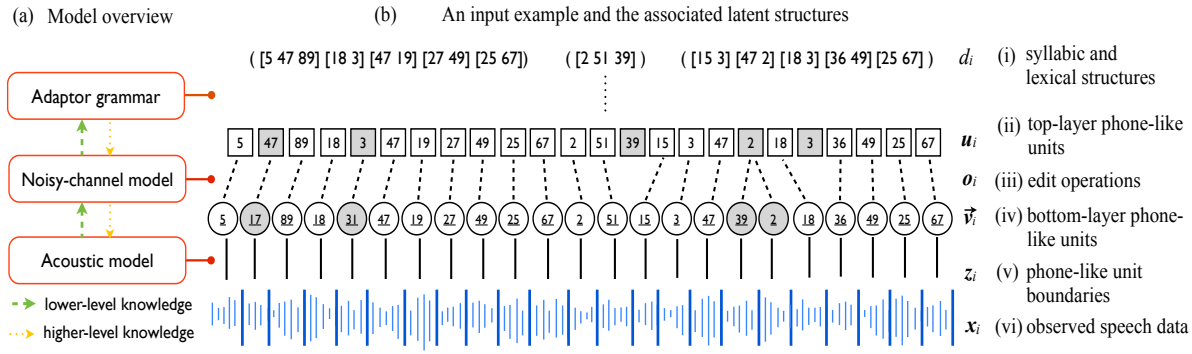


Figure 3.2. (a) An overview of the proposed model for inducing hierarchical linguistic structures directly from acoustic signals. The model consists of three major components: an adaptor grammar, a noisy-channel model, and an acoustic model. As indicated in the graph, the learning framework for the model allows partial knowledge learned from each level to drive discovery in the others. (b) An illustration of an input example, $\boldsymbol{x}_i$, and the associated latent structures in the acoustic signals $d_i, \boldsymbol{u}_i, \boldsymbol{o}_i, \vec{\boldsymbol{v}}_i, \boldsymbol{z}_i$. These latent structures can each be discovered by one of the three components of the model as specified by the red horizontal bars between (a) and (b). See Section 3.3 for a more detailed description.

blue bars in Fig. 3.2-(vi), which correspond to the boundaries of the speech segments. Each speech segment is labelled with a PLU id $v_{i,j,k} \in \mathbb{L}$, in which $\mathbb{L}$ is a set of integers that represent the PLU inventory embedded in the speech corpus. We denote the sequence of PLU ids associated with utterance i using $\vec{\boldsymbol{v}}_i$ as shown in Fig. 3.2-(iv), where $\vec{\boldsymbol{v}}_i = \{\boldsymbol{v}_{i,j} | 1 \leq j \leq J_i\}$ and $\boldsymbol{v}_{i,j} = \{v_{i,j,k} | v_{i,j,k} \in \mathbb{L}, 1 \leq k \leq |\boldsymbol{v}_{i,j}|\}$. The variable J_i is defined in the discussion of the second sub-task.

As depicted in Fig. 3.2-(a), we construct an acoustic model to approach this sub-problem. More specifically, the acoustic model is composed of a set of Hidden Markov Models (HMMs), $\boldsymbol{\pi}$, that are used to infer and model the PLU inventory from the given spoken utterances. With these HMMs, we can segment and decode the speech signals $\boldsymbol{x}_i$ into a series of PLUs $\vec{\boldsymbol{v}}_i$.

Phone variability modeling In conversational speech, phonetic realization of a word can easily vary because of the context, the stress patterns, etc. Take the word *the* as an example. While the pronunciation /dh ax/ is generally identified as the standard, the word can also be pronounced as /dh iy/ when it precedes a word that starts with a vowel, or when it is used to emphasize the proceeding noun. Without a mechanism that can map these two pronunciations into a unique representation, any model that induces linguistic structures based on phonetic input would fail to recognize these two pronunciations as instances of the same word type.

We exploit a noisy-channel model to address this problem and design three edit operations for the noisy-channel model: substitution, split, and deletion. Each of the operations takes a PLU as an input and is denoted as $\mathrm{sub}(u)$, $\mathrm{split}(u)$, and $\mathrm{del}(u)$ respectively. We assume that for every inferred sequence of PLUs $\vec{\boldsymbol{v}}_i$ in Fig. 3.2-(b)-(iv), there is a corresponding series of PLUs, $\boldsymbol{u}_i = \{u_{i,j} | 1 \leq j \leq J_i\}$, in which the pronunciations for any repeated word in $\vec{\boldsymbol{v}}_i$ are identical. The variable J_i indicates the length of $\boldsymbol{u}_i$. By passing each $u_{i,j}$ through the noisy-channel model, which stochastically chooses an edit operation $o_{i,j}$ for $u_{i,j}$, we obtain the noisy phonetic realization $\boldsymbol{v}_{i,j}$. Note that we denote $\boldsymbol{v}_{i,j}$ as a vector since if a split operation is chosen for $u_{i,j}$, the output of the noisy-channel model will contain two PLUs. The relationship among $\boldsymbol{u}_i$, $\boldsymbol{o}_i$, and $\vec{\boldsymbol{v}}_i$ is shown in (ii)-(iv) of Fig. 3.2-(b). The pairs of white squares and circles depict substitution operations in which $u_{i,j}$ is replaced with the same PLU. On the other hand, the pairs of shaded squares and circles illustrate the split and deletion operations along with examples of substitution operations where $u_{i,j}$ is substituted with a different PLU. In order to distinguish between the two sequences of PLUs $\boldsymbol{u}_i$ and $\vec{\boldsymbol{v}}_i$, we refer to the input of the noisy-channel model $\boldsymbol{u}_i$ as the top-layer PLUs and the output $\vec{\boldsymbol{v}}_i$ as the bottom-layer PLUs.

Syllabic and lexical unit learning With the standardized phonetic representation $\boldsymbol{u}_i$ obtained, higher-level linguistic structures such as the syllabic and lexical units can be inferred for each spoken utterance. We employ an Adaptor Grammar (AG) [83] to achieve the goal as indicated in Fig. 3.2-(a) and use d_i to denote the parse tree that encodes the hierarchical linguistic structures as shown in Fig. 3.2-(b)-(i).

In summary, we integrate an adaptor grammar with a noisy-channel model and an acoustic model to achieve the goal of discovering hierarchical linguistic structures directly from acoustic signals. Even though the sub-tasks are discussed in a bottom-up manner, our model provides a joint learning framework, allowing knowledge learned from one sub-task to drive discovery in the others as illustrated in Fig. 3.2-(a). In the rest of this section, we provide a review on AGs and then formally define the noisy-channel and acoustic model. At the end, we present the generative process implied by our model.

■ 3.3.2 Adaptor Grammars

Adaptor grammars are a non-parametric Bayesian extension of the Probabilistic Context-Free Grammars (PCFGs). In this section, we briefly review the definition of PCFGs and then show how to extend PCFGs to AGs. We refer readers to [83] and Chapter 3 of [140] for a more detailed description of AGs and their connection to PCFGs.

A PCFG can be defined as a quintuple $(N, T, R, S, \{\vec{\theta}^q\}_{q\in N})$, which consists of disjoint finite sets of nonterminal symbols N and terminal symbols T, a finite set of production rules $R \subseteq \{N \rightarrow (N \cup T)^*\}$, a start symbol $S \in N$, and vectors of probabilistic distributions $\{\vec{\theta}^q\}_{q\in N}$. In particular, each $\vec{\theta}^q$ contains the probabilities associated with the rules that have the nonterminal q on their left-hand side, which are denoted as the rule set R^q. We use θ_r to indicate the probability of rule $r \in R$. In our implementation, we adopt a Bayesian learning framework and impose a Dirichlet prior on each $\vec{\theta}^q \sim Dir(\vec{\alpha}^q)$, where $\vec{\alpha}^q$ represents the vector of hyperparameters for the Dirichlet distribution associated with q.

We use t to denote a *complete derivation*, which represents either a tree that expands from a nonterminal node q to its leaves, which contain only terminal symbols, or a tree that is composed of a single terminal symbol. We define $\mathrm{root}(t)$ as a function that returns the root node of t and denote the k immediate subtrees of the root node as $\hat{t}_1, \cdots, \hat{t}_k$. The probability distribution over $\mathcal{T}^q$, the set of trees that have $q \in N \cup T$ as the root, is recursively defined as follows.

$$G^q_{\text{pcfg}}(t) = \begin{cases} \sum_{r\in R^q} \theta_r \prod_{i=1}^{k} G^{\mathrm{root}(\hat{t}_i)}_{\text{pcfg}}(\hat{t}_i) & \mathrm{root}(t) = q \in N \\ 1 & \mathrm{root}(t) = q \in T \end{cases} \tag{3.1}$$

Eq. 3.1 completes the definition of a PCFG, which can be extended to an adaptor grammar. Simply put, an adaptor grammar is a sextuple $(N, T, R, S, \{\vec{\theta}^q\}_{q\in N}, \{Y^q\}_{q\in N})$, in which $(N, T, R, S, \{\vec{\theta}^q\}_{q\in N})$ is a PCFG, and $\{Y^q\}_{q\in N}$ is a set of *adaptors* for the nonterminals. An adaptor Y^q is a function that maps a base distribution over $\mathcal{T}^q$ to *a distribution* over distributions over $\mathcal{T}^q$. The distribution $G^q_{\text{ag}}(t)$ for $q \in N$ of an AG is a sample from *this distribution over distributions*. More specifically,

$$G^q_{\text{ag}}(t) \sim Y^q(H^q(t)) \tag{3.2}$$

$$H^q(t) = \sum\nolimits_{r\in R^q} \theta_r \prod\nolimits_{i=1}^{k} G^{\mathrm{root}(\hat{t}_i)}_{\text{ag}}(\hat{t}_i) \tag{3.3}$$

where $H^q(t)$ denotes the base distribution over $\mathcal{T}^q$. In this chapter, we use adaptors that are based on Pitman-Yor processes [150]; therefore, each Y^q corresponds to two parameters $\langle a^q, b^q\rangle$, which are the hyperparameters of the Pitman-Yor process. We use $\boldsymbol{a} = \{a^q | q \in N\}$ and $\boldsymbol{b} = \{b^q | q \in N\}$ to represent the sets of hyperparameters associated with the nonterminal symbols. If $a^q = 1$, then the adaptor Y^q becomes an identity function and $G^q_{\text{ag}}(t) = H^q(t)$, which indicates that the nonterminal q is not adapted. For terminal symbols $q \in T$, we define

$G^q_{\text{ag}}(t) = 1$, which is a distribution that puts all its probability mass on the single-node tree labelled q.

Conceptually, AGs can be regarded as PCFGs with memories, which cache the complete derivations of adapted nonterminals in a grammar and allow AGs to either reuse the cached trees or select a production rule in R to expand an adapted nonterminal. We denote the set of trees that the AG caches for $q \in N$ in the parses $\boldsymbol{D}$ of a corpus as $\boldsymbol{X}^q(\boldsymbol{D})$ and use $\boldsymbol{n}^q(\boldsymbol{D})$ to list the number of times that each cached tree expands q in $\boldsymbol{D}$. Furthermore, we gather the top expansion rule for each tree in $\boldsymbol{X}^q(\boldsymbol{D})$ and assign this set of rules to $\boldsymbol{T}^q$. The frequency of each rule appearing at the top of the trees in $\boldsymbol{X}^q(\boldsymbol{D})$ is stored in $\boldsymbol{f}^q$, which implies that the sum of $\boldsymbol{f}^q$ equals the length of $\boldsymbol{X}^q(\boldsymbol{D})$.

Since our goal is to discover the latent hierarchical linguistic structures in spoken utterances, we model an utterance as a sequence of words, represent a word as a sequence of syllables, and view syllables as a series of phones. We encode this hierarchical structure in the following AG as a constraint for parsing each spoken utterance.

$$\begin{aligned} &\text{Sentence} \rightarrow \underline{\text{Word}}^+ \\ &\underline{\text{Word}} \rightarrow \underline{\text{Syllable}}^+ \\ &\underline{\text{Syllable}} \rightarrow \text{Phone}^+ \\ &\text{Phone} \rightarrow l \quad \text{for } l \in \mathbb{L} \end{aligned} \tag{3.4}$$

We adopt the notations of [82] and use underlines to indicate adapted nonterminals and employ $^+$ to abbreviate right-branching recursive rules for nonterminals. The last rule shows that the terminals of this AG are the PLU ids, which are represented as $\boldsymbol{u}_i$ and depicted as the units in the squares of Fig. 3.2-(b)-(ii).

■ 3.3.3 Noisy-channel Model

We formulate the noisy-channel model as a PCFG and encode the substitution, split, and deletion operations as the grammar rules. In particular, for $l \in \mathbb{L}$,

$$\begin{aligned} &l \rightarrow l' && \text{for } l' \in \mathbb{L} \\ &l \rightarrow 1'_1 l'_2 && \text{for } l'_1, l'_2 \in \mathbb{L} \\ &l \rightarrow \epsilon \end{aligned} \tag{3.5}$$

where $l \in \mathbb{L}$ are the start symbols as well as the nonterminals of the PCFG. The terminals of this PCFG are $l' \in \mathbb{L}$, which correspond to the bottom-layer PLUs $\vec{\boldsymbol{v}}_i$ that are depicted as the

units in circles in Fig. 3.2-(b)-(iv). Note that $\{l\}$ and $\{l'\}$ correspond to the same set of PLUs. However, we exploit two sets of notations to specify that $\{l'\}$ are the terminals of this grammar and cannot be further expanded. The three sets of rules respectively map to the $\mathrm{sub}(\cdot)$, $\mathrm{split}(\cdot)$, and $\mathrm{del}(\cdot)$ operations; thus, the probability of each edit operation is automatically captured by the corresponding rule probability. We impose a Dirichlet prior on the rule probability distribution associated with each nonterminal l. More specifically, we assume $\vec{\theta}^l \sim Dir(\vec{\alpha}^l)$, where $\vec{\alpha}^l$ is a $(|\mathbb{L}|^2 + |\mathbb{L}| + 1)$-dimensional vector, whose entries correspond to the pseudo counts for the $|\mathbb{L}|^2$ split rules, $|\mathbb{L}|$ substitution rules, and the 1 deletion rule. Note that the first rule set $l \rightarrow l'$ includes the identity substitution.

As far as the discussion is concerned, the size of the PLU inventory is assumed to be known. This is a reasonable assumption if the corpus consists of speech data in a language whose phonetic properties are well studied, such as English. However, if the corpus is in a language for which not much prior knowledge is available, then the size of the phonetic inventory embedded in the corpus will need to be inferred from the data. Here we formulate a noisy-channel model that simultaneously infers a phonetic inventory of an unknown size and models the phone variability by using the infinite PCFG [117]. More concretely, we define the following grammar rules.

$$\text{Phone} \rightarrow l \qquad \text{for } l \in \mathbb{L} \tag{3.6}$$

$$l \rightarrow l_{\mathrm{sub}} \mid l_{\mathrm{split}} \mid l_{\mathrm{del}} \tag{3.7}$$

$$l_{\mathrm{sub}} \rightarrow l' \qquad \text{for } l' \in \mathbb{L} \tag{3.8}$$

$$l_{\mathrm{split}} \rightarrow l'_1 l'_2 \qquad \text{for } l'_1, l'_2 \in \mathbb{L} \tag{3.9}$$

$$l_{\mathrm{del}} \rightarrow \epsilon \tag{3.10}$$

The last rule of the AG defined in Eq. 3.4 involves the PLUs to be discovered; therefore, we repeat it here in Eq. 3.6. The second rule shown in Eq. 3.7 specifies the three edit operations that can be applied to a top-layer PLU. The outcomes of applying each edit operation on a top-layer PLU are listed in Eq. 3.8-3.10, which correspond to the grammar rules shown in Eq. 3.5. Note that the size of $\mathbb{L}$ is unknown in this grammar. We impose the following priors on the probabilistic distributions associated with the nonterminal symbols in Eq. 3.6-3.10. These priors allow us to induce a PLU inventory of a proper size directly from the data.

$$\vec{\theta}^{\text{Phone}} \sim GEM(\alpha^{\text{Phone}}) \tag{3.11}$$

$$\vec{\theta}^{l} \sim Dir(\vec{\alpha}^{l}) \tag{3.12}$$

$$\vec{\theta}^{l_{\text{sub}}} \sim DP(\alpha^{l_{\text{sub}}}, \vec{\theta}^{\text{Phone}}) \tag{3.13}$$

$$\vec{\theta}^{l_{\text{split}}} \sim DP(\alpha^{l_{\text{split}}}, \vec{\theta}^{\text{Phone}}\vec{\theta}^{\text{Phone}^T}) \tag{3.14}$$

$$\vec{\theta}^{l_{\text{del}}} = \delta(\epsilon) \tag{3.15}$$

In particular, we let $\vec{\theta}^{\text{Phone}}$ distribute according to the stick-breaking distribution [166] as shown in Eq. 3.11, where α^{Phone} is the parameter of the Beta distribution used in the stick-breaking construction process. Each entry of $\vec{\theta}^{\text{Phone}}$ represents a phone-like unit that our model discovers. To ensure that the input $\{l\}$ and the output $\{l'\}$, $\{l'_1\}$, and $\{l'_2\}$ of the noisy-channel model in Eq. 3.6-3.9 correspond to the same set of PLUs, we impose DP priors on $\vec{\theta}^{l_{\text{sub}}}$ and $\vec{\theta}^{l_{\text{split}}}$ by using $\vec{\theta}^{\text{Phone}}$ to construct the base distributions as shown in Eq. 3.13 and Eq. 3.14. More specifically, $\vec{\theta}^{\text{Phone}}\vec{\theta}^{\text{Phone}^T}$ is the product distribution over pairs of PLUs, and $\alpha^{l_{\text{sub}}}$ and $\alpha^{l_{\text{split}}}$ are the concentration parameters of the DPs. The probabilistic distribution associated with the deletion operation of Eq. 3.10 is the one that concentrates all its probability mass on the symbol ϵ, which is denoted as $\delta(\epsilon)$ in Eq. 3.15. Finally, a 3-dimensional Dirichlet distribution is imposed on $\vec{\theta}^{l}$ as shown in Eq. 3.12, which indicates the prior probability of choosing each of the three edit operations.

For computational efficiency, we only explore the finite version of the noisy-channel model for the experiments reported in this chapter. In particular, we employ the DPHMM model introduced in Chapter 2 to infer the PLU inventory from the speech data first. We then build the noisy-channel model described in Eq. 3.5 based on the discovered PLUs. However, the infinite noisy-channel model defined in Eq. 3.11-3.15 demonstrates the extendability of our approach to a full non-parametric Bayesian framework. It is not clear whether the full non-parametric Bayesian model would perform better than its finite counterpart in practice. Nevertheless, with the flexibility in phonetic inventory learning, we regard the full non-parametric Bayesian model as a more cognitively plausible framework for capturing the early language acquisition process.

■ 3.3.4 Acoustic Model

Finally, we assign each discovered PLU $l \in \mathbb{L}$ with an HMM, π_l, which is used to model the speech realization of each phonetic unit in the feature space. In particular, to capture the temporal dynamics of the features associated with a PLU, each HMM contains three emission

states, which roughly correspond to the beginning, middle, and end of a phonetic unit [76]. We model the emission distribution of each state by using 39-dimensional diagonal Gaussian Mixture Models (GMMs). The prior distributions embedded in each of the HMMs are the same as those described in Section 2.5. Dirichlet prior is imposed on the transition probability distribution, and the mixture weights of the GMM, for each state. In addition, a normal-Gamma distribution is applied as the prior for each Gaussian component in the GMMs.

■ 3.3.5 Generative Process of the Proposed Model

With the adaptor grammar, the noisy-channel model, and the acoustic model defined, we summarize the generative process implied by our model as follows. For the i^{th} utterance in the corpus, our model

1. Generates a parse tree d_i from $G_{\text{ag}}^{\text{Sentence}}(d)$.
2. For each leaf node $u_{i,j}$ of d_i, samples an edit rule $o_{i,j}$ from $\vec{\theta}^{u_{i,j}}$ to convert $u_{i,j}$ to $\boldsymbol{v}_{i,j}$.
3. For $v_{i,j,k} \in \boldsymbol{v}_{i,j}$, $1 \leq k \leq |\boldsymbol{v}_{i,j}|$, generates the speech features using $\pi_{v_{i,j,k}}$, which deterministically sets the value of $z_{i,t}$.

The generative process explicitly points out the latent variables our model defines for each utterance, which we summarize as follows.

- d_i: the parse tree that encodes the hierarchical linguistic structures of the i^{th} training sample.
- $\boldsymbol{u}_i$: the top-layer PLUs.
- $\boldsymbol{o}_i$: the set of edit operations applied to $\boldsymbol{u}_i$.
- $\vec{\boldsymbol{v}}_i$: the bottom-layer PLUs.
- $\boldsymbol{z}_i$: the phonetic segmentation hidden in the acoustic signals.
- $\boldsymbol{\pi}$: the HMM parameters.
- $\{\vec{\theta}^q\}_{q \in N_{\text{ag}} \cup N_{\text{noisy-channel}}}$: the rule probabilities of the base PCFG in the adaptor grammar and of the noisy-channel model.

In the next section, we derive the inference methods for all the latent variables except for $\{\vec{\theta}^q\}_{q \in N_{\text{ag}} \cup N_{\text{noisy-channel}}}$, which we integrate out during the inference process.

■ 3.4 Inference

We exploit Markov chain Monte Carlo algorithms to generate samples from the posterior distribution over the latent variables. In particular, we construct three sampling steps to move on the Markov chain: 1) jointly sampling d_i, $\boldsymbol{o}_i$, $\boldsymbol{u}_i$, 2) generating new samples for $\boldsymbol{o}_i$, $\vec{\boldsymbol{v}}_i$, $\boldsymbol{z}_i$, and 3) updating $\boldsymbol{\pi}$. In the rest of this section, we describe each of the sampling moves in detail.

■ 3.4.1 Sampling d_i, $\boldsymbol{o}_i$, and implicitly $\boldsymbol{u}_i$

We employ the Metropolis-Hastings (MH) algorithm [19] to generate samples for d_i and $\boldsymbol{o}_i$, which implicitly determines $\boldsymbol{u}_i$. Given the current bottom-layer PLUs of the i^{th} utterance, $\vec{\boldsymbol{v}}_i$, and $d_{-i}, \boldsymbol{o}_{-i}$, which are the current parses and the current edit operations associated with all the sentences in the corpus except the i^{th} utterance, we construct the following proposal distribution for d_i' and $\boldsymbol{o}_i'$. We use d_i and d_i' to distinguish the current from the proposed parse. The relationship between $\boldsymbol{o}_i$ and $\boldsymbol{o}_i'$ is similarly defined. Note that this proposal distribution is an approximation of the true joint conditional posterior distribution of d_i and $\boldsymbol{o}_i$.

$$
\begin{aligned}
& p'(d_i', \boldsymbol{o}_i' | \vec{\boldsymbol{v}}_i, d_{-i}, \boldsymbol{o}_{-i}; \{\vec{\alpha}^q\}, \boldsymbol{a}, \boldsymbol{b}) && (3.16)\\
&= p'(d_i', \boldsymbol{o}_i' | d_{-i}, \boldsymbol{o}_{-i}; \{\vec{\alpha}^q\}, \boldsymbol{a}, \boldsymbol{b}) \qquad \text{for } \boldsymbol{o}_i' \in \{\boldsymbol{o}_i' | \mathrm{rhs}(\boldsymbol{o}_i') = \vec{\boldsymbol{v}}_i\} && (3.17)\\
&= p'(\boldsymbol{o}_i' | \boldsymbol{o}_{-i}; \{\vec{\alpha}\}^q) p'(d_i' | d_{-i}, \boldsymbol{o}_i'; \{\vec{\alpha}\}^q, \boldsymbol{a}, \boldsymbol{b}) && (3.18)\\
&\approx \underbrace{\prod_{o_{i,j}' \in \boldsymbol{o}_i'} \frac{C_{-i}(u_{i,j}' \to \boldsymbol{v}_{i,j}') + \vec{\alpha}^{u_{i,j}'}_{u_{i,j}' \to \boldsymbol{v}_{i,j}'}}{C_{-i}(u_{i,j}') + \sum_{r \in R^{u_{i,j}'}} \vec{\alpha}_r^{u_{i,j}'}}}_{\text{(a)}} \underbrace{p'(d_i' | d_{-i}, \boldsymbol{o}_i'; \{\alpha\}^q, \boldsymbol{a}, \boldsymbol{b})}_{\text{expanded by the approximating PCFG in [83]}} && (3.19)
\end{aligned}
$$

where $q \in N_{\text{ag}} \cup N_{\text{noisy-channel}}$. Since the current top-layer PLUs of all the other utterances $\boldsymbol{u}_{-i}$, the boundary variables $\boldsymbol{z}_i$ and $\boldsymbol{z}_{-i}$, the HMM parameters $\boldsymbol{\pi}$, and the observed speech signals $\boldsymbol{x}_i$, are independent of d_i' and $\boldsymbol{o}_i'$ given $\vec{\boldsymbol{v}}_i, d_{-i}, \boldsymbol{o}_{-i}$, we omit $\boldsymbol{u}_{-i}$, $\boldsymbol{z}_i$, $\boldsymbol{z}_{-i}$, $\boldsymbol{\pi}$, and $\boldsymbol{x}_i$ in Eq. 3.16. The function $\mathrm{rhs}(\boldsymbol{o}_i')$ returns the sequence of the right-hand side symbols of each edit operation in $\boldsymbol{o}_i'$. Therefore, Eq. 3.17 results from only considering the set of edit operation sequences $\boldsymbol{o}_i'$ whose right-hand side symbols match the given $\vec{\boldsymbol{v}}_i$. We apply the chain rule to Eq. 3.17 and use the conditional independence properties $\boldsymbol{o}_i' \perp\!\!\!\perp d_{-i} | \boldsymbol{o}_{-i}$ and $d_i' \perp\!\!\!\perp \boldsymbol{o}_{-i} | \boldsymbol{o}_i'$ implied by our model to obtain Eq. 3.18.

The notation $C_{-i}(w)$ in Eq. 3.19 denotes the number of times that w is used in the analyses for the corpus, excluding the i^{th} utterance, in which w can be any countable entity such as a rule

or a symbol. We estimate $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i};\{\vec{\alpha}\}^q)$ in Eq. 3.18 by multiplying the relative frequency of each edit operation $o'_{i,j}$ in $\boldsymbol{o}'_i$ using counts $C_{-i}(u'_{i,j} \rightarrow \boldsymbol{v}'_{i,j})$ and $C_{-i}(u'_{i,j})$ as well as $\vec{\alpha}^{u'_{i,j}}$ as shown in Eq. 3.19. Note that Eq. 3.19-(a) is only an approximation for $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i};\{\vec{\alpha}\}^q)$ since the counts $C_{-i}(u'_{i,j} \rightarrow \boldsymbol{v}'_{i,j})$ and $C_{-i}(u'_{i,j})$ are not incremented according to $\boldsymbol{o}'_i$. More clearly, if any edit operation appears more than once in $\boldsymbol{o}'_i$, then Eq. 3.19-(a) will not contain the exact value of $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i};\{\vec{\alpha}\}^q)$. As for the second term of Eq. 3.18, we employ the approximating PCFGs for AGs described in [83] to estimate the probability of every possible d'_i as indicated by the last term of Eq. 3.19.

Eq. 3.19 defines the joint probability of d'_i and $\boldsymbol{o}'_i$ under the proposal distribution. To efficiently compute the probabilities of all possible combinations of d'_i and $\boldsymbol{o}'_i$, which is required for drawing a sample from the proposal distribution, we construct a new PCFG G'. In particular, we combine the PCFG that approximates the adaptor grammar with the PCFG whose rule set consists of all the edit operations $o'_{i,j}$ used and weighted as in Eq. 3.19-(a). The new PCFG G' is thus a grammar that can be used to parse the terminals $\vec{\boldsymbol{v}}_i$ and generate derivations that are rooted in the start symbol of the AG. More specifically, with G', we can transform the task of sampling d'_i and $\boldsymbol{o}'_i$ from the proposal distribution to the task of generating a parse for $\vec{\boldsymbol{v}}_i$ using G'. The latter task can be efficiently solved by using an adaptation of the Inside-Outside algorithm for PCFGs [109], which is described in detail by [84], [63], and [43].

Once new proposals for d'_i and $\boldsymbol{o}'_i$ are generated, we can accept the proposals with probability $A(\boldsymbol{d},\boldsymbol{o},\boldsymbol{d}',\boldsymbol{o}')$ shown in Eq. 3.20, where $\boldsymbol{d}' = \{d_{-i}, d'_i\}, \boldsymbol{o}' = \{\boldsymbol{o}_{-i}, \boldsymbol{o}'_i\}$, and $\boldsymbol{d}$ is the same as $\boldsymbol{d}'$ except that d'_i is replaced with d_i. The set of edit operations $\boldsymbol{o}$ is defined in a similar manner.

$$A(\boldsymbol{d},\boldsymbol{o},\boldsymbol{d}',\boldsymbol{o}') = \min\{1, \frac{p_{\text{model}}(\boldsymbol{d}',\boldsymbol{o}'|\boldsymbol{z},\boldsymbol{x};\{\vec{\alpha^q}\},\boldsymbol{a},\boldsymbol{b})p'(d_i,\boldsymbol{o}_i|\boldsymbol{v}_i,d_{-i},\boldsymbol{o}_{-i};\{\vec{\alpha}^q\},\boldsymbol{a},\boldsymbol{b})}{p_{\text{model}}(\boldsymbol{d},\boldsymbol{o}|\boldsymbol{z},\boldsymbol{x};\{\vec{\alpha}^q\},\boldsymbol{a},\boldsymbol{b})p'(d'_i,\boldsymbol{o}'_i|\boldsymbol{v}_i,d_{-i},\boldsymbol{o}_{-i};\{\vec{\alpha}^q\},\boldsymbol{a},\boldsymbol{b})}\} \tag{3.20}$$

where $p_{\text{model}}(\boldsymbol{d},\boldsymbol{o}|\boldsymbol{z},\boldsymbol{x};\{\vec{\alpha^q}\},\boldsymbol{a},\boldsymbol{b})$ is the joint posterior probability of $\boldsymbol{d}$ and $\boldsymbol{o}$ defined by our model, which can be decomposed into $p_{\text{ag}}(\boldsymbol{d}|\boldsymbol{o};\{\vec{\alpha^q}\},\boldsymbol{a},\boldsymbol{b})p_{\text{noisy-channel}}(\boldsymbol{o};\{\vec{\alpha^q}\})$ as follows.

$$p_{\text{ag}}(\boldsymbol{d}|\boldsymbol{o};\{\vec{\alpha^q}\},\boldsymbol{a},\boldsymbol{b}) = \prod_{q\in N_{\text{ag}}} \underbrace{\frac{\prod_{k=1}^{|\boldsymbol{X}^q(\boldsymbol{d})|}(a^q(k-1)+b^q)\prod_{j=1}^{\boldsymbol{n}_k^q(\boldsymbol{d})-1}(j-1-a^q)}{\prod_{i=1}^{\text{sum}(\boldsymbol{n}^q(\boldsymbol{d}))}(i-1+b^q)}}_{\text{(a)}} \underbrace{\frac{\prod_{k=1}^{|\boldsymbol{T}^q|}\prod_{j=1}^{\boldsymbol{f}_k^q}(j-1+\vec{\alpha}^q_{\boldsymbol{T}_k^q})}{\prod_{i=1}^{\text{sum}(\boldsymbol{f}^q)}(i-1+\sum_{r\in R^q}\vec{\alpha}^q_r)}}_{\text{(b)}} \tag{3.21}$$

where $\boldsymbol{X}^q(\boldsymbol{d})$, $\boldsymbol{n}^q(\boldsymbol{d})$, $\boldsymbol{T}^q$ and $\boldsymbol{f}^q$ are defined in Section 3.3.2. Eq. 3.21-(a) is the probability of using the cached subtrees to expand q in $\boldsymbol{d}$ according to the Pitman-Yor process associated with q, with H^q in Eq. 3.3 integrated out. Eq. 3.21-(b) is the probability of choosing the top rule in each cached subtree, computed by integrating over $\vec{\theta}^q, q \in N_{\text{ag}}$. Similarly, by integrating out $\vec{\theta}^q, q \in N_{\text{noisy-channel}}$, we can obtain

$$p_{\text{noisy-channel}}(\boldsymbol{o}; \{\vec{\alpha}^q\}) = \prod_{q \in N_{\text{noisy-channel}}} \frac{\prod_{k=1}^{|\boldsymbol{E}^q(\boldsymbol{o})|} \prod_{j=1}^{\boldsymbol{c}_k^q(\boldsymbol{o})} (j - 1 + \vec{\alpha}^q_{\boldsymbol{E}_k^q(\boldsymbol{o})})}{\prod_{i=1}^{\text{sum}(\boldsymbol{c}^q(\boldsymbol{o}))} (i - 1 + \sum_{r \in R^q} \vec{\alpha}^q_r)} \tag{3.22}$$

where $\boldsymbol{E}^q(\boldsymbol{o})$ represents the set of edit operation rules used in $\boldsymbol{o}$ that have q on their left-hand side, and $\boldsymbol{c}^q(\boldsymbol{o})$ contains the number of times that each of the rules in $E^q(\boldsymbol{o})$ appears in $\boldsymbol{o}$.

■ 3.4.2 Sampling $\boldsymbol{z}_i$, $\boldsymbol{o}_i$, and implicitly $\vec{\boldsymbol{v}}_i$

The d_i and $\boldsymbol{o}_i$ obtained in Section 3.4.1 deterministically assign a new sequence of top-layer PLUs $\boldsymbol{u}_i$ to the i^{th} utterance. Given the new $\boldsymbol{u}_i$ and the speech data $\boldsymbol{x}_i$, we describe how to generate new samples for the boundary variables $\boldsymbol{z}_i$ and the bottom-layer PLUs $\vec{\boldsymbol{v}}_i$. Similar to the previous sampling move, updating $\vec{\boldsymbol{v}}_i$ given $\boldsymbol{u}_i$ is equivalent to replacing the edit operation sequence $\boldsymbol{o}_i$ with a new $\boldsymbol{o}'_i$ that satisfies the constraint that $\text{lhs}(\boldsymbol{o}'_i) = \boldsymbol{u}_i$. In particular, $\text{lhs}(\boldsymbol{o}'_i)$ is a function that returns the sequence of the left-hand side symbols of each edit rule in $\boldsymbol{o}'_i$. By taking this viewpoint, we can transform the problem of updating $\boldsymbol{z}_i$ and $\vec{\boldsymbol{v}}_i$ to the task of sampling new values for $\boldsymbol{z}_i$ and $\boldsymbol{o}_i$. In this section, we present the inference method by using the two sets of variables $\{\boldsymbol{z}_i, \boldsymbol{o}_i\}$ and $\{\boldsymbol{z}_i, \vec{\boldsymbol{v}}_i\}$ interchangeably.

We develop a Metropolis-Hastings sampler and construct the following proposal distribution $p'(\boldsymbol{o}'_i, \boldsymbol{z}'_i | \boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$, which approximates the true conditional posterior of $\boldsymbol{o}_i$ and $\boldsymbol{z}_i$ defined by our model.

$$p'(\boldsymbol{o}'_i, \boldsymbol{z}'_i | \boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\}) = p'(\boldsymbol{o}'_i, \boldsymbol{z}'_i | \boldsymbol{x}_i, \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\}) \quad \text{for } \boldsymbol{o}'_i \in \{\boldsymbol{o}'_i | \text{lhs}(\boldsymbol{o}'_i) = \boldsymbol{u}_i\} \tag{3.23}$$

$$= \underbrace{\frac{p'(\boldsymbol{o}'_i, \boldsymbol{z}'_i, \boldsymbol{x}_i | \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}{\sum_{\boldsymbol{o}'_i, \boldsymbol{z}'_i} p'(\boldsymbol{o}'_i, \boldsymbol{z}'_i, \boldsymbol{x}_i | \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}}_{\text{(a)}} = \underbrace{\frac{p'(\boldsymbol{o}'_i | \boldsymbol{o}_{-i}; \{\vec{\alpha}^q\}) p'(\boldsymbol{z}'_i, \boldsymbol{x}_i | \boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}{\sum_{\boldsymbol{o}'_i, \boldsymbol{z}'_i} p'(\boldsymbol{o}'_i | \boldsymbol{o}_{-i}; \{\vec{\alpha}^q\}) p'(\boldsymbol{z}'_i, \boldsymbol{x}_i | \boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}}_{\text{(b)}} \tag{3.24}$$

where $q \in N_{\text{noisy-channel}}$. Eq. 3.23 reflects the constraint we impose on $\boldsymbol{o}'_i$. We apply Bayes' rule to Eq. 3.23 to reach Eq. 3.24-(a). Furthermore, the chain rule and the conditional independence properties $\boldsymbol{o}'_i \perp\!\!\!\perp \boldsymbol{\pi}|\boldsymbol{o}_{-i}$ and $\boldsymbol{x}_i, \boldsymbol{z}'_i \perp\!\!\!\perp \boldsymbol{o}_{-i}|\boldsymbol{o}'_i$ are exploited to decompose $p(\boldsymbol{o}'_i, \boldsymbol{z}'_i, \boldsymbol{x}_i|\boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ of Eq. 3.24-(a) into $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i}; \{\vec{\alpha}^q\})p'(\boldsymbol{z}'_i, \boldsymbol{x}_i|\boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ in Eq. 3.24-(b).

The product $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i}; \{\vec{\alpha}^q\})p'(\boldsymbol{z}'_i, \boldsymbol{x}_i|\boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ is approximated under the proposal distribution by Eq. 3.25.

$$p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i}; \{\vec{\alpha}^q\})p'(\boldsymbol{z}'_i, \boldsymbol{x}_i|\boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\}) \approx \underbrace{\prod_{o'_{i,j} \in \boldsymbol{o}'_i} \frac{C_{-i}(u'_{i,j} \rightarrow \boldsymbol{v}'_{i,j}) + \vec{\alpha}^{u'_{i,j}}_{u'_{i,j} \rightarrow \boldsymbol{v}'_{i,j}}}{C_{-i}(u'_{i,j}) + \sum_{r \in R^{u'_{i,j}}} \vec{\alpha}^{u'_{i,j}}_r}}_{\text{defined to be } Q(\boldsymbol{o}'_i)} \prod_{v'_{i,j,k} \in \boldsymbol{v}'_{i,j}, \boldsymbol{v}'_{i,j} \in \vec{\boldsymbol{v}}'_i} p(\boldsymbol{x}_{i,j,k}|\pi_{v'_{i,j,k}}) \tag{3.25}$$

where we estimate the probability of each edit rule $o'_{i,j} \in \boldsymbol{o}'_i$ by its relative frequency observed in $\boldsymbol{o}_{-i}$. As shown in Eq. 3.25, $p'(\boldsymbol{o}'_i|\boldsymbol{o}_{-i}; \{\vec{\alpha}^q\})$ is approximated by the product of the relative frequency of each rule $o'_{i,j}$ in $\boldsymbol{o}'_i$. The boundary variables $\boldsymbol{z}'_i$ deterministically split the speech data $\boldsymbol{x}_i$ into a sequence of speech segments as illustrated by the dark blue bars in Fig. 3.2-(vi). Each of the speech segments is linked to a bottom-layer PLU $v_{i,j,k}$, for which we denote the speech segment as $\boldsymbol{x}_{i,j,k}$. In addition, as shown in Fig. 3.2, each of the speech segments is further mapped to a top-layer PLU through the edit operations. The last term of Eq. 3.25 represents the emission probability of $\boldsymbol{x}_{i,j,k}$ given the corresponding HMM $\pi_{v'_{i,j,k}}$, which can be efficiently computed with the forward-backward algorithm. Note that different segmentations correspond to different sets of $\boldsymbol{x}_{i,j,k}$. Therefore, $p(\boldsymbol{o}'_i|\boldsymbol{o}_{-i}; \{\vec{\alpha}^q\})p(\boldsymbol{z}'_i, \boldsymbol{x}_i|\boldsymbol{o}'_i, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ in Eq. 3.25 is a function of not only $\boldsymbol{o}'_i$ but also the speech segmentation encoded by $\boldsymbol{z}'_i$.

Eq. 3.25 specifies the joint probability of $\boldsymbol{x}_i$, $\boldsymbol{z}'_i$ and $\boldsymbol{o}'_i$ under the proposal distribution. To generate proposals for $\boldsymbol{o}'_i$ and $\boldsymbol{z}'_i$, we need to compute the joint probability for each $\boldsymbol{o}'_i \in \{\boldsymbol{o}'_i|\text{lhs}(\boldsymbol{o}'_i) = \boldsymbol{u}_i\}$ combined with all the possible segmentations encoded by $\boldsymbol{z}'_i$. We modify the efficient message-passing algorithm for hidden semi-Markov models [132] to achieve the goal. To derive our algorithm, we define and compute the backwards messages B and B^*. First,

$$\begin{aligned} B_t(j) &\triangleq p(x_{i,t+1:T_i}|\mathbb{I}(x_{i,t}) = j, z_{i,t} = 1) \\ &= \sum_{m=j+1}^{J_i} B^*_t(m) \prod_{n=j+1}^{m-1} \theta'_{u_{i,n} \rightarrow \epsilon} \end{aligned} \tag{3.26}$$

where $\mathbb{I}(x_{i,t})$ is a function that returns the index of the top-layer PLU that $x_{i,t}$ is mapped to, and $x_{i,t_1:t_2}$ is an abbreviation that represents the speech segment consisting of speech features $x_{i,t_1}, \cdots, x_{i,t_2}$. By definition, $B_t(j)$ is the probability of $x_{i,t+1:T_i}$, which can be calculated by summing over all possible segmentations in $x_{i,t+1:T_i}$ and all the valid alignments between $u_{i,j+1:J_i}$ and $x_{i,t+1:T_i}$. Note that we use $u_{i,j_i:j_2}$ to denote the partial sequence of the top-layer PLUs $u_{i,j_1}, \cdots, u_{i,j_2}$. More specifically, the value of $B_t(j)$ can be computed recursively by using B^* as shown in Eq. 3.26, which is defined as follows.

$$B_t^*(j) \triangleq p(x_{i,t+1:T_i}|\mathbb{I}(x_{i,t+1}) = j, z_{i,t} = 1) \tag{3.27}$$

$$= \sum_{m=1}^{T_i-t} \underbrace{p(x_{i,t+1:t+m}|u_{i,j})}_{\text{(a)}} B_{t+m}(j) \tag{3.28}$$

$$\begin{aligned} &= \sum_{m=1}^{T_i-t} \{\sum_{l' \in \mathbb{L}} \theta'_{u_{i,j} \to l'} p(x_{i,t+1:t+m}|\pi_{l'}) \\ &+ \sum_{l'_1 \in \mathbb{L}} \sum_{l'_2 \in \mathbb{L}} \theta'_{u_{i,j} \to l'_1 l'_2} p(x_{i,t+1:t+m}|\pi_{l'_1}\pi_{l'_2})\} B_{t+m}(j) \end{aligned} \tag{3.29}$$

$$\begin{aligned} &= \sum_{m=1}^{T_i-t} \{\sum_{l' \in \mathbb{L}} \theta'_{u_{i,j} \to l'} p(x_{i,t+1:t+m}|\pi_{l'}) \\ &+ \sum_{n=1}^{m-1} \sum_{l'_1 \in \mathbb{L}} \sum_{l'_2 \in \mathbb{L}} \theta'_{u_{i,j} \to l'_1 l'_2} p(x_{i,t+1:t+n}|\pi_{l'_1}) p(x_{i,t+n+1:t+m}|\pi_{l'_2})\} B_{t+m}(j) \end{aligned} \tag{3.30}$$

$B_t^*(j)$ is the probability of $x_{i,t+1:T_i}$ given that the speech segment starting with $x_{i,t+1}$ is mapped to $u_{i,j}$, which is explicitly expressed in Eq. 3.28-(a). The summation in Eq. 3.28 runs through all possible durations for the segment beginning at $x_{i,t+1}$, and $B_{t+m}(j)$ stands for the total probability of $x_{i,t+m+1:T_i}$. The probability $p(x_{i,t+1:t+m}|u_{i,j})$ can be obtained by marginalizing $p(\boldsymbol{v}_{i,j}|u_{i,j})p(x_{i,t+1:t+m}|\boldsymbol{\pi}, \boldsymbol{v}_{i,j})$ over all $\boldsymbol{v}_{i,j}$ that $u_{i,j}$ can map to based on $R^{u_{i,j}}$. In particular, we exploit the substitution and split edit rules in $R^{u_{i,j}}$, and use the probabilities associated with these rules, $\theta'_{u_{i,j} \to l'}$ and $\theta'_{u_{i,j} \to l'_1 l'_2}$, to compute $p(x_{i,t+1:t+m}|u_{i,j})$ as demonstrated in Eq. 3.29. Specifically, the probability θ'_r is estimated by the relative frequency of r with the counts in $\boldsymbol{o}_{-i}$ as described in Eq. 3.25. The two items $p(x_{i,t+1:t+m}|\pi_{l'})$ and $p(x_{i,t+1:t+m}|\pi_{l'_1}\pi_{l'_2})$ in Eq. 3.29 are the likelihood of observing $x_{i,t+1:t+m}$ given the HMMs that are associated with the bottom-layer PLUs l' and $l'_1 l'_2$. Finally, we expand $p(x_{i,t+1:t+m}|\pi_{l'_1}\pi_{l'_2})$ in Eq. 3.30 to explicitly list all possible positions of the boundary between the two segments that are mapped to l'_1 and l'_2. The initialization and the termination steps for computing B and

B^* are shown in Eq 3.31-3.32 and Eq. 3.33 respectively.

$$\text{(Initialization)} \quad B_{T_i}(j) \triangleq \begin{cases} 1 & \text{if } j = J_i \\ \prod_{m=j+1}^{J_i} \theta_{u_{i,m} \to \epsilon} & \text{if } j < J_i \end{cases} \tag{3.31}$$

$$\text{(Initialization)} \quad B_t(J_i) \triangleq \begin{cases} 1 & \text{if } t = T_i \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

$$\text{(Termination)} \quad B_0(0) = \sum_{j=1}^{J_i} B_0^*(j) \prod_{m=1}^{j-1} \theta_{u_{i,m} \to \epsilon} \tag{3.33}$$

where Eq. 3.32 specifies that no feature vectors can be left unaligned with the top-layer PLUs. The message-passing algorithm ends when the value of $B_0(0)$ is computed. By definition, $B_0(0)$ carries the sum of the probabilities of all possible segmentations in $x_{i,1:T_i}$ (encoded in $\boldsymbol{z}_i'$) and the alignments between $x_{i,1:T_i}$ and $\boldsymbol{u}_{i,1:J_i}$ (captured by $\vec{\boldsymbol{v}}_i'$).

With B and B^*, we can generate samples of $\vec{\boldsymbol{v}}_i'$ and $\boldsymbol{z}_i'$ from the proposal distribution using the forward-sampling scheme presented in Alg. 3.4.1. The function $SampleFromB_t(j)$ in line 6 generates a sample by using the relative probability associated with each entry of the summation defining B that is shown in Eq. 3.26 and Eq. 3.33. The generated sample represents the index of the top-layer PLU that is to be linked with the speech segment starting with $x_{i,t+1}$. If a top-layer PLU is not aligned with any speech features, which can be verified by using the criterion of line 7, then the ϵ symbol is assigned to the corresponding bottom-layer PLU as indicated by line 9 of Alg. 3.4.1.

The function $SampleFromB_t^*(next_j)$ in line 13 returns a tuple $\langle m, \boldsymbol{v}'_{i,next_j}, n \rangle$ by sampling from the normalized distribution composed of the summation entries of Eq. 3.30. The value m indicates the duration of the speech segment that starts from $x_{i,t+1}$. The vector $\boldsymbol{v}'_{i,next_j}$ contains the bottom-layer PLUs that $u_{i,next_j}$ maps to. If a split edit rule is sampled, then n specifies the location of the boundary between the two speech segments that are tied to $v_{i,next_j,1}$ and $v_{i,next_j,2}$. Implicitly, m and n determine the phone boundaries in the speech data and set the values of $\boldsymbol{z}_i'$ as shown in line 15 and line 17 of Alg. 3.4.1.

When the forward-sampling algorithm terminates, a new segmentation $\boldsymbol{z}_i'$ and a proposal of $\vec{\boldsymbol{v}}_i'$ for the i^{th} utterance are obtained. By combining $\boldsymbol{u}_i$ and $\vec{\boldsymbol{v}}_i'$, we can get a new sequence of edit operations $\boldsymbol{o}_i'$. The generated proposals $\boldsymbol{z}_i'$ and $\boldsymbol{o}_i'$ are then accepted with probability $A(\boldsymbol{z}, \boldsymbol{o}, \boldsymbol{z}', \boldsymbol{o}')$ defined in Eq. 3.34.

Algorithm 3.4.1 Generate proposals for $\vec{\boldsymbol{v}}'_i$ and $\boldsymbol{z}'_i$ from $B_t(j)$ and $B^*_t(j)$

$\vec{\boldsymbol{v}}'_i = [\,]$ **% initialize $\vec{v}'_i$ to be an empty array**
$\boldsymbol{z}'_i \leftarrow \mathbf{0}$ **% assign 0 to each entry of $\boldsymbol{z}'_i$**
$j \leftarrow 0$
$t \leftarrow 0$
while $j < J_i \wedge t < T_i$ **do**
    $next_j \leftarrow SampleFromB_t(j)$
    if $next_j > j + 1$ **then**
        for $k = j + 1$ to $k = next_j - 1$ **do**
            $\boldsymbol{v}'_{i,k} \leftarrow \epsilon$
            $\vec{\boldsymbol{v}}'_i$.append($\boldsymbol{v}'_{i,k}$) **% append $v'_{i,k}$ to $\vec{v}'_i$**
        end for
    else
        $\langle m, \boldsymbol{v}'_{i,next_j}, n\rangle \leftarrow SampleFromB^*_t(next_j)$
        $\vec{\boldsymbol{v}}'_i$.append($\boldsymbol{v}'_{i,next_j}$) **% append $v'_{i,next_j}$ to $\vec{v}'_i$**
        $z'_{i,t+m} \leftarrow 1$
        if $n \neq$ NULL **then**
            $z'_{i,t+n} \leftarrow 1$
        end if
    end if
    $t \leftarrow t + m$
    $j \leftarrow next_j$
end while

$$A(\boldsymbol{z}, \boldsymbol{o}, \boldsymbol{z}', \boldsymbol{o}') = \min\{1, \frac{p_{\text{model}}(\boldsymbol{z}', \boldsymbol{o}'|\boldsymbol{x}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})p'(\boldsymbol{z}_i, \boldsymbol{o}_i|\boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}{p_{\text{model}}(\boldsymbol{z}, \boldsymbol{o}|\boldsymbol{x}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})p'(\boldsymbol{z}'_i, \boldsymbol{o}'_i|\boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{o}_{-i}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}\} \tag{3.34}$$

where $\boldsymbol{z}$ comprised $\boldsymbol{z}_i$ (the current segmentation in $\boldsymbol{x}_i$) and $\boldsymbol{z}_{-i}$, which denotes the segmentations of all the utterances in the corpus except the i^{th} utterance. The notation $\boldsymbol{z}'$ is defined in a similar manner except that $\boldsymbol{z}_i$ is substituted for $\boldsymbol{z}'_i$. The probability $p_{\text{model}}(\boldsymbol{z}, \boldsymbol{o}|\boldsymbol{x}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ can be computed in the following way.

$$p_{\text{model}}(\boldsymbol{z}, \boldsymbol{o}|\boldsymbol{x}, \boldsymbol{\pi}; \{\vec{\alpha}\}^q) = \frac{p(\boldsymbol{z}, \boldsymbol{o}, \boldsymbol{x}|\boldsymbol{\pi}; \{\vec{\alpha}\}^q)}{\sum_{\underline{\boldsymbol{z}}} \sum_{\underline{\boldsymbol{o}}} p(\underline{\boldsymbol{z}}, \underline{\boldsymbol{o}}, \boldsymbol{x}|\boldsymbol{\pi}; \{\vec{\alpha}\}^q)} \quad \text{(Bayes' rule)}$$
$$= \frac{p_{\text{noisy-channel}}(\boldsymbol{o}; \{\vec{\alpha}^q\}) p(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{o}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})}{\sum_{\underline{\boldsymbol{z}}} \sum_{\underline{\boldsymbol{o}}} p(\underline{\boldsymbol{z}}, \underline{\boldsymbol{o}}, \boldsymbol{x}|\boldsymbol{\pi}; \{\vec{\alpha}\}^q)} \tag{3.35}$$

where $p_{\text{noisy-channel}}(\boldsymbol{o}; \{\vec{\alpha}\}^q)$ is defined in Eq. 3.22. It can be shown that the denominator and $p(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{o}, \boldsymbol{\pi}; \{\vec{\alpha}^q\})$ in Eq. 3.35 are cancelled out in $A(\boldsymbol{z}, \boldsymbol{o}, \boldsymbol{z}', \boldsymbol{o}')$. As a result, $A(\boldsymbol{z}, \boldsymbol{o}, \boldsymbol{z}', \boldsymbol{o}')$ can be simply reduced to $\frac{p_{\text{noisy-channel}}(\boldsymbol{o}'; \{\vec{\alpha}\}^q) Q(\boldsymbol{o}_i)}{p_{\text{noisy-channel}}(\boldsymbol{o}; \{\vec{\alpha}\}^q) Q(\boldsymbol{o}'_i)}$ using the $Q(\cdot)$ defined in Eq. 3.25.

■ 3.4.3 Sampling $\boldsymbol{\pi}$

Given $\boldsymbol{z}_i$ and $\vec{\boldsymbol{v}}_i$ of each utterance in the corpus, generating new samples for the parameters of each HMM π_l for $l \in \mathbb{L}$ is straightforward. We briefly summarize the process in this section and refer readers to Section 5 of Chapter 2 for a more detailed description. For each PLU l, we gather all speech segments that are mapped to a bottom-layer PLU $v_{i,j,k} = l$. For every segment in this set, we use π_l to block-sample the state id and the GMM mixture id for each feature vector. From the state and mixture assignments, we can collect the counts that are needed to update the priors for the transition probability and the emission distribution of each state in π_l. New samples for the parameters of π_l can thus be yielded from the updated priors.

■ 3.4.4 Parameters for the Model

We discuss the choices for the parameters of the adaptor grammar, the noisy-channel model and the acoustic model in this section.

Adaptor grammar The adaptor grammar has two sets of parameters: $\{\vec{\alpha}^q\}_{q \in N_{ag}}$, the parameters for the Dirichlet priors imposed on the rule probabilities in the base PCFG, and $\{a^q, b^q\}_{q \in N_{ag}}$, the hyperparameters for the Pitman-Yor processes. We let $\vec{\alpha}^q$ be an array of all ones for $q \in N_{ag}$. With this choice of $\vec{\alpha}^q$, we impose a weak prior on the rule probabilities $\vec{\theta}^q$ and favor neither sparse nor dispersed probability distributions for $\vec{\theta}^q$ a priori. As for a^q and b^q, rather than specifying a particular value for each of them, we apply a uniform $Beta(1, 1)$ prior on a^q and a vague $Gamma(10, 0.1)$ prior on b^q for $q \in \mathbb{L}$. The values of a^q and b^q are sampled after every sweep through the corpus for the inference steps described in the previous three sections. We exploit the publicly available software [79] that is implemented based on [83] to generate samples for a^q and b^q.

Noisy-channel model The parameters of the noisy-channel model are $\{\vec{\alpha}^q\}_{q \in N_{\text{noisy-channel}}}$ of the Dirichlet priors imposed on the probabilistic distributions over the edit operation rules. In particular, each $\vec{\alpha}^q$ has $|\mathbb{L}|^2 + |\mathbb{L}| + 1$ entries, which represent the prior preferences for applying the split, substitution, and deletion operations on the top-layer PLU q. To provide a strong learning constraint, we encourage the noisy-channel model to substitute a top-layer PLU with the exact matching bottom-layer PLU. More clearly, we put a large prior on the edit operation rule $l \rightarrow l'$ for $l' = l$. In practice, we set $\vec{\alpha}^q_{l \rightarrow l'}$ to be 2000 and the rest of the entries of $\vec{\alpha}^q$ to be 1. This choice of $\vec{\alpha}^q$ may seem extreme at first glance given that it is heavily biased towards to the matching edit operation. However, with a closer look, we can see that for a PLU inventory of 50 units, the marginal prior probability on the exact matching rule is only $\frac{2000}{2500+2000+49+1} \sim 0.44$, which can go lower when the size of the PLU inventory grows.

Acoustic model We set up the HMMs $\boldsymbol{\pi}$ the same way as in Section 2.4. Hence, the parameters used for the acoustic model can be retrieved from Table 2.2.

■ 3.5 Experimental Setup

In this section, we describe the dataset and the evaluation methods used to assess the effectiveness of the proposed model for discovering linguistic structures directly from acoustic signals. In addition, in order to study the importance of each component of the model, we construct a series of ablative systems by removing one component of the model at a time. We also describe the lesioned systems in this section.

■ 3.5.1 Dataset

To the best of our knowledge, there are no standard copora for evaluating models for automatic linguistic structure discovery. More specifically, a variety of datasets have been used in previous work, including a child-directed speech corpus [33, 34, 82], the MIT Lecture corpus [146, 191], the WSJCAM0 corpus of read news articles [69], the Switchboard corpus of short telephone conversations [75], and a corpus of Mandarin broadcast news [20].

In this paper, we perform experiments on the same six lecture recordings used in [146, 191], which are a part of the MIT Lecture corpus [59]. A brief summary of the six lectures is listed in Table 3.1. The reason for testing our model on the MIT Lecture corpus is threefold. First, compared to the broadcast news corpus and the read speech corpus used in [20] and [69], the speaking style observed in the lecture data is more spontaneous, which creates a more challenging learning task for our model. Second, each of the six lectures consists of speech

Lecture topic	Speaker	Duration
Economics	Thomas Friedman	75 mins
Speech processing	Victor Zue	85 mins
Clustering	James Glass	78 mins
Speaker adaptation	Timothy Hazen	74 mins
Physics	Walter Lewin	51 mins
Linear algebra	Gilbert Strang	47 mins

Table 3.1. A brief summary of the six lectures used for the experiments reported in Section 3.6.

data from a different speaker with a duration of up to one hour or more. Unlike the telephone conversation corpus used in [75], this characteristic of the Lecture corpus allows us to train the model on a set of single-speaker speech data. This can significantly reduce the amount of noise in the experimental results that may arise from the problem of speaker variability. Finally, the learning of our model is contingent on the existence of repeating patterns in the data. Therefore, in order to gain useful insights into our model, the testing speech data must contain recurrent patterns that allow our model to learn. Since each lecture is about a well-defined topic, it often contains a set of highly frequent subject-specific words. These subject-specific words correspond to an abundance of repeated acoustic patterns in the speech data that our model can leverage for learning.

■ 3.5.2 Systems

In this section, we describe the various systems that are compared in the experiments, which include the full model and two lesioned ones. Also, we explain how to initialize the training for each of the systems.

Full system Two full systems based on the model described in Sec. 3.3 are constructed. The only difference between the two systems is how the value of K, the size of the PLU inventory, is determined. We fix the value of K to be 50 for one system, and for the other system, we let the value of K be discovered by the DPHMM framework presented in Chapter 2 for each lecture. The number of PLUs that the DPHMM model finds for each lecture is shown in Table 3.2. We refer to these two systems as Full50 and FullDP respectively.

Initialization The training of the FullDP system is initialized by using the output of the DPHMM model for each lecture. More specifically, as shown in Chapter 2, the DPHMM

Lecture	# units
Economics	99
Speech processing	111
Clustering	91
Speaker adaptation	83
Physics	90
Linear algebra	79

Table 3.2. Number of phonetic units found by DPHMM for each lecture.

model learns a set of HMMs, discovers the phone boundaries in the acoustic signals, and assigns a PLU label to each speech segment. We exploit the HMMs, the boundaries, and the PLU ids found by the DPHMM model as the initial values for the latent variables $\boldsymbol{\pi}$, $\boldsymbol{b}_i$, and $\vec{\boldsymbol{v}}_i$ of the FullDP system. After initialization, the training of FullDP proceeds by following the three sampling moves described in Section 3.4. Similarly, we employ a Hierarchical HMM (HHMM), which is presented in detail in Section 4.3, to find the initial values of $\boldsymbol{\pi}$, $\boldsymbol{b}_i$, and $\vec{\boldsymbol{v}}_i$ for the Full50 system. In addition to the full systems, the lesioned systems that are described in the rest of this section are also initialized in the same manner.

No acoustic model We remove the acoustic model from Full50 and FullDP to obtain the first lesioned systems, which are denoted as the -AM (read as minus AM) systems. For example, Full50-AM represents the Full50 system without the acoustic model. Since the lesioned systems do not have an acoustic model, they can neither resegment nor relabel the speech data. In other words, the initial HMMs, the segmentation, and the bottom-layer PLUs remain intact during training for the -AM systems. This also implies that there is no learning of phonetic units in the -AM systems. Therefore, by comparing a -AM system to its full counterpart, we can investigate the synergies between phonetic and lexical unit acquisition in the full model.

No noisy-channel To evaluate the importance of modeling phonetic variability, we further remove the noisy-channel module from the -AM systems to form the -NC systems. More specifically, the -NC systems treat the initial bottom-layer PLUs $\vec{\boldsymbol{v}}_i$ as the top-layer PLUs $\boldsymbol{u}_i$ and keep $\boldsymbol{u}_i$ the same for the entire training period. A -NC system can thus be regarded as a pipeline framework for discovering linguistic structures from speech, in which the phone sequence of each utterance is discovered as the first step, and the latent linguistic structures are learned in the second step.

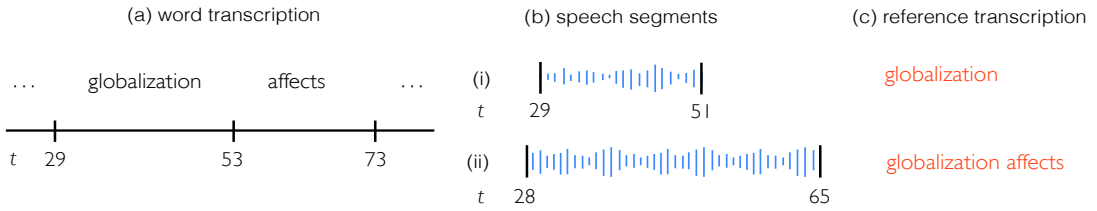


Figure 3.3. An illustration of how the reference transcriptions of the speech segments discovered by our model are determined: (a) partial forced-aligned word transcription used for illustration, (b) examples of speech segments that our model may discover for the sentence, and (c) the reference transcription for each of the example speech segments determined by using the procedure described in Section 3.5.3. The t variable indicates the time indices of the boundaries of the words in the transcription and those of the speech segments.

■ 3.5.3 Evaluation methods

Two evaluation metrics are used to quantitatively gauge the effectiveness of the systems described above for discovering linguistic structures from acoustic signals. We describe the two metrics in this section.

Coverage of words with high TFIDF scores As mentioned before, each lecture contains a set of frequent subject-specific words. Therefore, to assess the quality of the lexical units our model discovers, we test whether the induced lexical units correspond to these subject-specific vocabulary. In particular, we employ the top 20 highest TFIDF scoring words as the target words to be learned for each lecture. We compare the coverage achieved by our model to that obtained by the baseline [146] and by the state-of-the-art system on this task [190].

Since each lexical unit induced by our model is abstracted as a sequence of PLUs, we adopt the following procedure to identify the *word label* for each lexical unit. Specifically, for each discovered lexical unit of each lecture, we first determine the reference transcription of each speech segment that is associated with the lexical unit. To find the reference transcription of a speech segment, we search the forced-aligned word transcriptions of the lecture to find the word w_s that has the closest starting time to that of the speech segment. Similarly, we look for the word w_e, whose ending time is the closest to that of the speech segment. The word sequence spanning from w_s to w_e is then assigned as the reference transcription to the speech segment. Fig. 3.3 shows two examples of speech segments along with their corresponding reference transcriptions. Having found the reference transcription for each segment associated with the lexical unit, we then choose the word or the phrase that appears most frequently as the label. Lexical units with no majority word or phrase are not assigned with any labels.

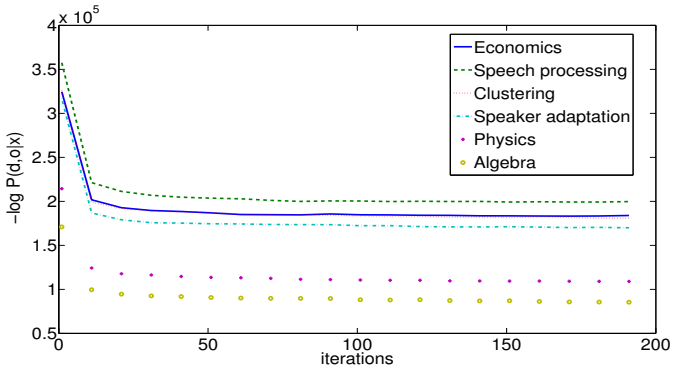


Figure 3.4. The negative log posterior probability of the latent variables $\boldsymbol{d}$ and $\boldsymbol{o}$ as a function of iteration obtained by the **Full50** system for each lecture.

Phone segmentation Besides the coverage of high TFIDF words, we also evaluate our model on the task of phone segmentation for the six lectures. We use a speech recognizer to produce phone forced alignments for each lecture. The phone segmentation embedded in the forced alignments is then treated as the *gold standard* to which we compare the segmentation our model generates. We follow the suggestion of [165] and use a 20-ms tolerance window to compute the F1 score of the phone segmentation discovered by our model. Because the segmentations of the -AM and -NC systems are identical to those obtained by the initialization systems, we only compare the F1 scores of the full systems and those achieved by the -AM systems.

■ 3.6 Results and Analysis

Before presenting the model's performance on the tasks of coverage of target words and phone segmentation, we briefly discuss some qualitative behaviors of the model.

Training convergence Figs. 3.4-3.6 show the negative log posterior probability of the sampled parses $\boldsymbol{d}$ and edit operations $\boldsymbol{o}$ (except for the Full50-NC system) for each lecture as a function of iteration generated by Full50, Full50-AM, and Full50-NC. Given that each lecture consists of roughly only one hour of speech data, we can see that the Full50(-AM, -NC) systems all converge fairly quickly within just a couple hundreds of iterations. The FullDP(-AM, -NC) systems also demonstrate similar convergence behaviors. In particular, the negative log posterior probability of $\boldsymbol{d}$ and $\boldsymbol{o}$ for the FullDP system is shown in Fig. 3.7. In the this section, we report the performance of each system using the corresponding sample from the 200^{th}

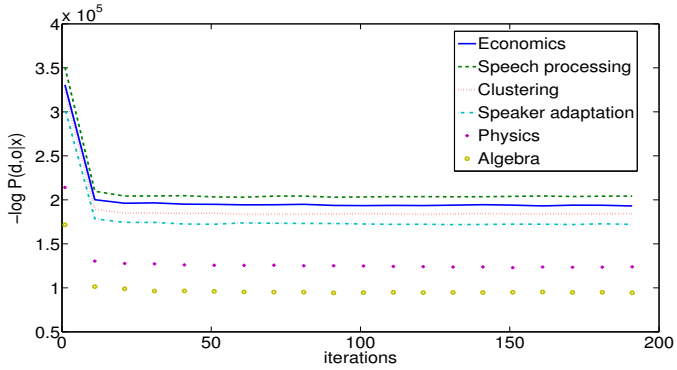


Figure 3.5. The negative log posterior probability of the latent variables $\boldsymbol{d}$ and $\boldsymbol{o}$ as a function of iteration obtained by the **Full50-AM** system for each lecture.

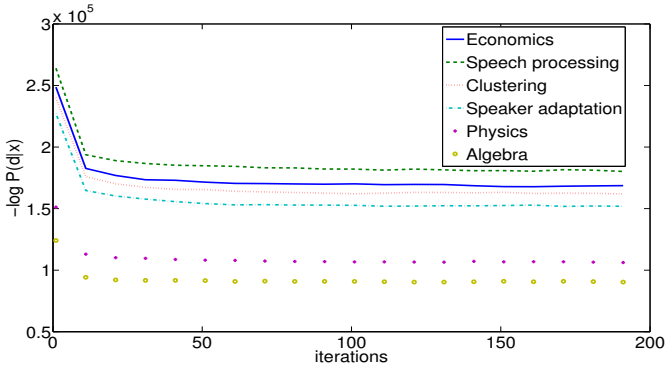


Figure 3.6. The negative log posterior probability of the latent variables $\boldsymbol{d}$ as a function of iteration obtained by the **Full50-NC** system for each lecture.

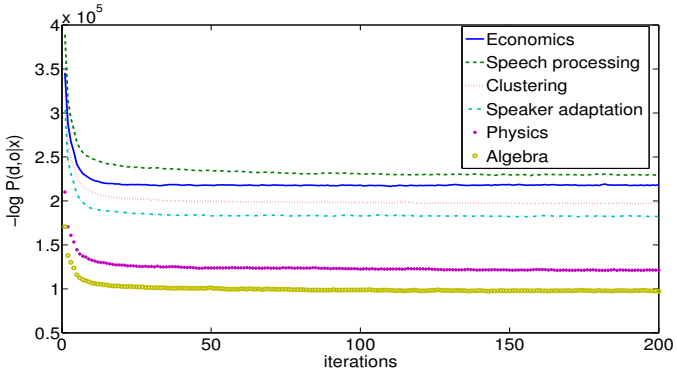


Figure 3.7. The negative log posterior probability of the latent variables $\boldsymbol{d}$ and $\boldsymbol{o}$ as a function of iteration obtained by the **FullDP** model for each lecture.

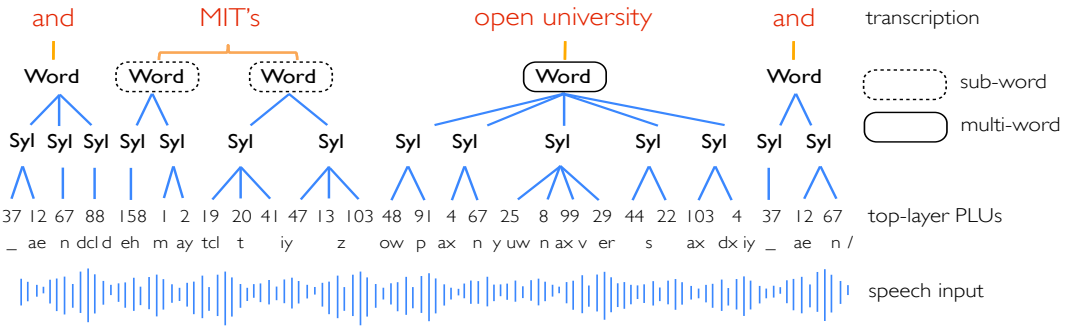


Figure 3.8. The parse our model generates for the sentence "*and MIT's open university and*," an utterance excerpted from the economics lecture.

iteration.

Analysis of the discovered word units Unlike previous methods for discovering lexical units from acoustic signals [146, 190, 75], which can only find isolated speech segments scattered throughout utterances in a dataset, our model aims to learn the continuous sequence of words underlying each sentence. We show the word sequence, the syllable sequence, and the top-layer PLUs that the FullDP system learns for the sentence "*and MIT's open university and*," extracted from the economics lecture, in Fig. 3.8. The phone labels of the top-layer PLUs are listed for illustration. Fig. 3.8 shows that our model successfully recognizes the word "and" at the beginning and the end of the sentence as two word tokens, while it generates a single lexical unit for the phrase "open university" and maps two sub-word lexical units to the word "MIT's." The parse shown in Fig. 3.8 reveals the first observation we make about our model's learning behavior: the model learns lexical units that correspond to *sub-words*, *single words*, and *multi-word phrases*.

We list a subset of the lexical units discovered by the FullDP system for the economics lecture in Table 3.3. Each of the lexical units is represented by its underlying syllabic structures, which are denoted in $[\cdot]$ and shown in the second column. The transcription and the number of speech segments, $|\text{Word}|$, that are associated with each lexical unit are also shown in the first and the third columns of the table. For lexical units that correspond to sub-words, we label the phone sequences for the sub-words and list some examples of the words in the dataset that our model parses using these sub-word lexical units. Table 3.3 shows that the discovered lexical units with high frequencies, or large $|\text{Word}|$s, are usually sub-words that can be used to parse many different words. As the frequency decreases, the corresponding lexical unit starts to map to single words and eventually multi-word phrases.

Transcription	Discovered lexical units	|Word|
/iy l iy/ (really, willy, billion)	[35] [31 4]	68
/ah_fp m/ (<um>, company)	[105 46 8] [14]	51
/f ao r m/ (form, informing)	[81] [39 139 8]	49
/w ah n/ (one, want, wonderful)	[172] [79 71]	49
/ey sh ax n/ (innovation, imagination)	[6 7 30] [49]	43
/_ ae n d/ (and, end, attend)	[37] [12 67]	41
/f l ae t/ (flat, flatten)	[81] [28 16 21]	39
/ih z/ (this, is, it's)	[47 59]	33
the world	[35 40 102] [38 91]	31
/ax bcl ax l/ (able, cable, incredible)	[34 18] [38 91]	18
/aw n dh/ (down, how, download)	[11] [95 8]	16
/s cl t eh r/ (this era, first era)	[70 110 3] [25 5]	15
/s iy/ (c, seen, seeing)	[59] [41]	14
together	[15 26 25] [27 99]	13
china	[19 7] [151 2]	10
discovered	[26] [70 110 3] [9 99] [31]	9
people	[126 20] [15 14] [39 38]	9
you know	[37 25] [27 48 91]	8
globalization	[106 48] [18 31] [147 13] [6 7 30]	7
individual	[49 146] [34 99] [154] [54 7] [35 48]	7
because	[37] [15 50] [106 27 13]	7
two thousand	[19 20 25] [52] [9] [13 173]	6
global	[160] [106 48] [38 18] [38 91]	6
convergen(ce/ed)	[7 30] [54] [18 29] [45 7 30]	5
powerful	[50 57 145] [145] [81 39 38]	5
collaboration	[50 137] [28 16] [18 31 43] [6 7 30]	5
new form of	[1 35] [39 139 8] [48 91]	5
quiet	[50 106] [172 24] [50 106 43]	4

Continued on next page

Table 3.3 – continued from previous page

Transcription	Discovered lexical units	|Word|
southwest airlines	[70 59] [48 91] [106 32] [70 110 3] [5 40 79 2]	4
open university	[48 91] [4 67] [25 8 99 29] [44 22] [103 4]	4
for multiple forms	[39 139 8] [38 91] [106 98] [39 139 8]	4
field	[154] [54] [52] [25] [35 48]	4
miles away	[13 173] [1 2] [70 59 103] [40 32]	4
the world is	[99 35 40 139] [38 91] [47 13]	3
steam engines	[70 110 3] [41] [67 6] [54 20 30 1]	3
it seems to be	[47 59] [41] [70 59 103] [67] [4] [99 77]	3
and she said	[37 12] [54 7] [70 59 103]	3
the new york	[34 99] [158 71] [25 5] [19 57 95]	3
first chapter	[10 29 23] [7 30] [11 21] [5]	2
the arab muslim world	[28 32] [41] [67] [25 35] [1 27] [13 173] [8 139] [38 91]	2
platform	[34 18] [27 21] [11 21] [137 16 21 105 139]	2

Table 3.3: A subset of the lexical units that the FullDP system discovers for the economics lecture. The number of independent speech segments that are associated with each lexical unit is denoted as |Word| and shown in the last column.

At first glance, Fig. 3.8 and Table 3.3 seem to suggest that our model is not capturing the right lexical structures because it learns many *sub-word* and *multi-word* lexical units. However, a closer look at the data reveals that this learning behavior is contingent on the frequencies of the observed acoustic patterns. Take the fifth lexical unit [6 7 30] in Table 3.3 that corresponds to the sound sequence /ey sh ax n/ as an example. Many words that contain this sound sequence are observed in the data, such as *conversation*, *reservation*, *innovation*, and *foundation*. However, most of these words only appear in the lecture a few times. For example, the frequencies of

the four words are only 2, 3, 4, and 2 times respectively. Therefore, from the model's point of view, it only sees recurrent acoustic patterns that map to the sound sequence /ey sh ax n/, but does not receive enough evidence for each of the words individually, which causes the model to pick up just /ey sh ax n/ as a word. Nonetheless, our model does acquire single-word lexical units for words that appear frequently. For example, two other words that also contain the sound sequence /ey sh ax n/ are *globalization* and *collaboration*, which occur 25 and 21 times respectively in the lecture. As shown in Table 3.3, our model is able to properly recognize and create a lexical unit for each of the two words.

Our analysis so far suggests that pattern frequency is the key to our model learning lexical units. However, a more careful examination of the frequencies of the multi-word phrases that our model captures discloses that there may be one more important factor that affects the learning behavior of our model, which is the *length* of the repeated acoustic patterns. For example, although the phrase *open university* only appears 5 times in the lecture, our model still captures the pattern and learns a lexical unit for it. The driving force behind this learning behavior may be that lexical units mapping to multi-word phrases tend to have a strong *parsing power*, which we roughly define as how much data a lexical unit can explain. For instance, a sub-word lexical unit should have a weaker parsing power than a multi-word lexical unit because the portion of an utterance it can parse is usually smaller than that which can be parsed by a multi-word lexical unit. Since our model has the tendency to acquire lexical units with a strong parsing power, the model can learn long repeated acoustic patterns even when the patterns only appear infrequently.

To be more concrete, imagine that our model needs to parse the following sequence of top-layer PLUs for the sentence, "*open university*" from the economics lecture, which, for clarity, is denoted by standard phone units: /ow p ax n y uw n ax v er s ax dx iy/. Assume that the model has already seen the same sequence of PLUs once and cached the rule: Word $\rightarrow$ /ow p ax n y uw n ax v er s ax dx iy/. Also, assume that the model has learned and reused the following sub-word lexical units 200 times each: Word $\rightarrow$ /ow p/, Word $\rightarrow$ /ax n/, Word $\rightarrow$ /y uw/, Word $\rightarrow$ /n ax v er/, Word $\rightarrow$ /s ax dx iy/. The two parsing choices that our model has are: 1) h_1, reuse the multi-word lexical unit and 2) h_0, reuse the four sub-word lexical units. For simplicity, we omit the Words $\rightarrow$ Word grammar rule in the AG for parsing, exclude the possibility of generating parses from the base PCFG, and focus on reusing the rules cached by the AG. Nevertheless, the logic behind the explanation remains the same when the parses from the PCFG are taken into account. The conditional posterior probabilities $p(h_1|\cdots)$ and $p(h_0|\cdots)$ our model assigns to the two hypotheses are shown as follows.

$$p(h_1|\cdots) \sim \frac{C_{-i}(\text{Word} \rightarrow \text{/ow p ax n y uw n ax v er s ax dx iy/})}{C_{-i}(\text{Word})} = \frac{1}{10,000} = 10^{-4} \quad (3.36)$$

$$p(h_0|\cdots) \sim \frac{C_{-i}(\text{Word} \rightarrow \text{/ow p/})}{C_{-i}(\text{Word})} \frac{C_{-i}(\text{Word} \rightarrow \text{/ax n/})}{C_{-i}(\text{Word})} \frac{C_{-i}(\text{Word} \rightarrow \text{/y uw/})}{C_{-i}(\text{Word})} \quad (3.37)$$
$$\times \frac{C_{-i}(\text{Word} \rightarrow \text{/n ax v er/})}{C_{-i}(\text{Word})} \frac{C_{-i}(\text{Word} \rightarrow \text{/s ax dx iy/})}{C_{-i}(\text{Word})} = (\frac{200}{10,000})^5 = 3.2^{-9}$$

where we approximate the number of word tokens that our model produces for other sentences in the lecture, $C_{-i}(\text{Word})$, to be $10,000$ (in fact, our model generates a total of $10,578$ word tokens for the economics lecture; therefore, $10,000$ is a quite close approximation). By comparing Eq. 3.36 and Eq. 3.37, we can see that because of the weak parsing power possessed by the sub-word lexical units, the model needs to utilize multiple sub-word lexical units to parse the sentence. Since the product in Eq. 3.37 involves many items, the posterior probability for h_0 drops quickly. On the contrary, as the multi-word lexical unit has a strong parsing power and can explain the sentence all by itself, the posterior probability for h_1 involves only one item and is much higher than that for h_0. The large difference between $p(h_1|\cdots)$ and $p(h_0|\cdots)$ drives the model to reuse the multi-word lexical unit even though the model has only seen the same sequence once before.

This analysis can also help us understand why our model requires more examples to learn lexical units for single words. The derivation of the analysis remains the same except that for single words, the product in Eq. 3.37 will involve fewer items, such as two, and thus yield a higher $p(h_0|\cdots) \sim 4 \times 10^{-4}$. In this case, unless the model has already seen the same word at least four times before and thus generates a $p(h_1|\cdots) \sim 4 \times 10^{-4}$, the model would prefer h_0 and reuse sub-word lexical units to parse the word. We will discuss this learning preference over structure reuse in more detail later in this section.

Lastly, to complete the discussion of our model's learning behavior for discovering lexical units with different granularities, we analyze the word sequence the FullDP system generates for each sentence of each lecture, and compute the proportions of the discovered lexical units that map to *sub-words*, *single words*, and *multi-word phrases*. The results, presented in Fig. 3.9, show that while a large portion of the discovered word tokens correspond to *single words*, many of the lexical units represent *sub-words* or *multi-word phrases*, which matches with our previous observation.

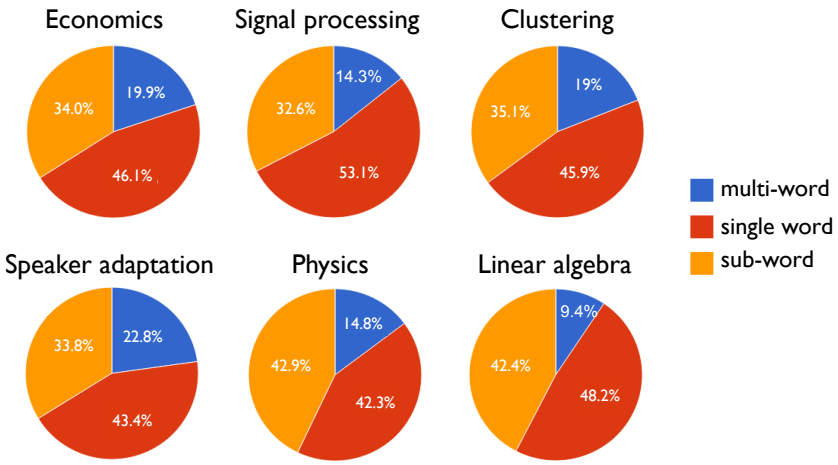


Figure 3.9. The proportions of the word tokens the FullDP system generates for each lecture that map to *sub-words*, *single words*, and *multi-words*.

Analysis of the discovered syllable units Table 3.4 lists a subset of the syllable structures that the FullDP system discovers for the economics lecture. Each of the syllable units is denoted by its underlying PLU sequence. The phone transcription and the number of speech segments associated with each syllable structure, |Syl|, are also shown in the table. For the transcription, we use () to indicate optional phones and | to specify various valid phone transcriptions. For example, the transcription /(ax|ah) s cl t|k/ in the third line of Table 3.4 indicates that the syllable unit 70 110 3 maps to a fricative /s/, followed by a stop /cl t/ or /cl k/, and some times the sequence of /s cl t/ or /s cl k/ is preceded by a vowel /ax/ or /ah/.

The definition of the notation used for the phone transcriptions reveals one observation we make from the discovered syllable structures: a syllable unit may correspond to various sound sequences. However, as shown in Table 3.4, the set of pronunciations that a syllable unit maps to usually consists of similarly sounding patterns. Besides the syllable unit 70, 110, 3 discussed previously, the syllable unit 11 21 in line six of Table 3.4 is shown to mostly map to /ae cl p/, /ae cl t/, and /ae cl k/, which are sound sequences consisting of the vowel /ae/, followed by a stop consonant. Secondly, Table 3.4 also shows that the discovered syllable units may not always match the standard definition of a syllable. Nonetheless, even though these learned syllable-like units do not perfectly map to standard ones, they often map to structures that are highly reusable for constructing different word types, as shown by the number of speech segments that are associated with each of the discovered syllable-like structures.

Transcription	Discovered syllabic units	|Syl|
_	37	587
/(iy) s eh/	70 59 103	348
/(ax|ah) s cl t|k/	70 110 3	346
/s|z ax|ih/	13 173	294
/(ah) l (vcl d|cl t)/	38 91	205
/ae cl p|t|k)/	11 21	201
/n|m/	67	195
/f/	81	168
/ih/	154	167
/ow|uh cl/	48 91	166
/n vcl|cl/	54	161
/iy/	41	137
/ey sh ax/	6 7 30	126
/ae n (dcl)/	12 67	105
/(d) ax v/	34 99	98
/th|f/	52	88
/(f) ao r m/	39 139 8	74
/g l ow/	106 48	62
/ax gcl g (eh)/	15 26 25	62
/v er|bcl ax r/	18 29	62
/ih (z|s)/	47 59	51
/m ao/	8 139	50
/ow/	35 48	48
/f el|ao/	81 39 38	44
/t|k ax gcl/	34 18	33
/bcl el/	18 98	31
/p l/	137 32	26
/cl ch/	19 7	20
/dh ah w er l dcl/	99 35 40 139	18
/eh r l ay/	5 40 79 2	15

Continued on next page

Table 3.4 – continued from previous page

Transcription	Discovered syllabic units	|Syl|
/y uw/	147 115	8
/k ah z/	106 27 13	7

Table 3.4: A subset of the syllabic units that the FullDP system infers from the economics lecture. The value |Syl| specifies the number of speech segments that are labeled with each syllable structure.

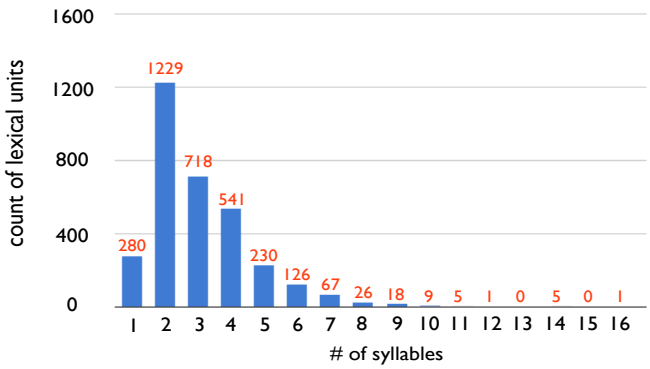


Figure 3.10. The distribution of the number of syllables contained in a discovered lexical unit.

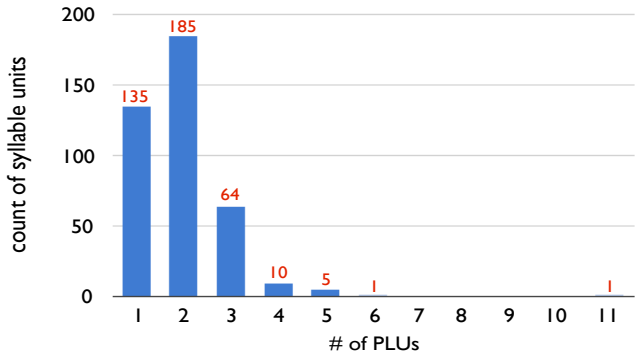


Figure 3.11. The distribution of the number of top-layer PLUs underlying a discovered syllable structure.

Table 3.3 and Table 3.4 also illustrate that our model discovers lexical units that are composed of syllable units, and learns syllable units that are composed of top-layer PLUs. This

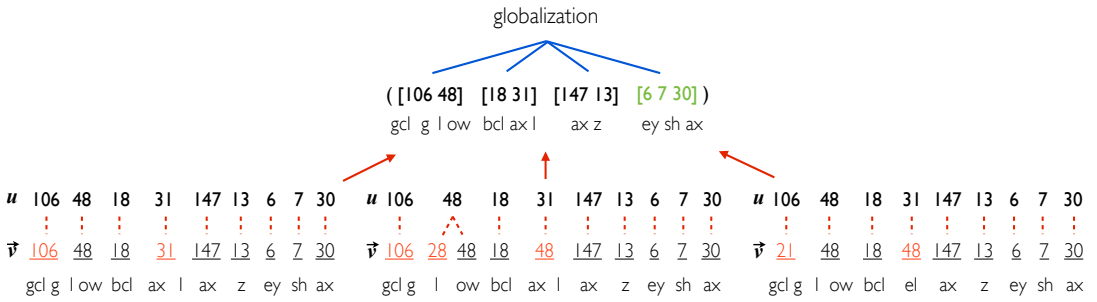


Figure 3.12. The bottom-layer PLUs $\vec{v}$ and the top-layer PLUs $\boldsymbol{u}$ as well as the syllable structures that the FullDP system discovers for three spoken examples of the word *globalization*. The phonetic and syllabic structures are denoted with phone transcriptions for clarity.

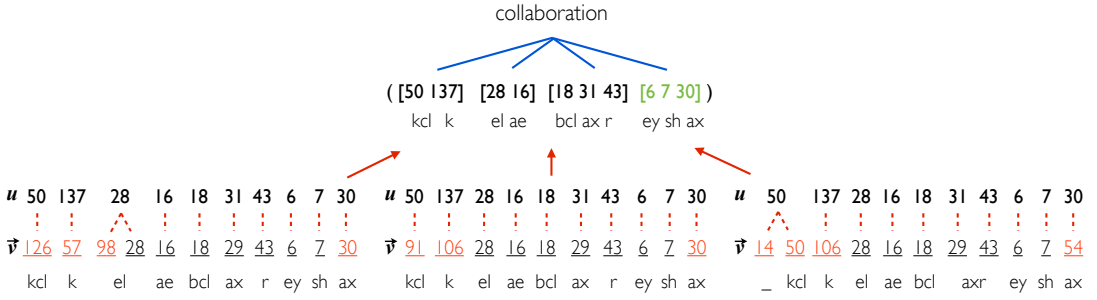


Figure 3.13. The bottom-layer PLUs $\vec{v}$ and the top-layer PLUs $\boldsymbol{u}$ as well as the syllable structures that the FullDP system discovers for three spoken examples of the word *collaboration*. The phonetic and syllabic structures are denoted with phone transcriptions for clarity.

observation suggests that the model is indeed inferring the latent linguistic structures based on the given grammar. To gain more insight, we plot the distribution of the number of syllable units contained in each discovered lexical unit in Fig. 3.10 and the distribution of the counts of PLUs underlying the syllabic structures in Fig. 3.11. From Fig. 3.10, we can see that most of the lexical units consist of two or more syllables. As indicated in Table 3.3, the lexical units containing fewer than two syllables tend to correspond to sub-words, and those that comprise more syllabic units map to single words or multi-word phrases. Similarly, Fig. 3.11 shows that most of the discovered syllable units contain more than two PLUs, which usually correspond to a combination of vowels and consonants as shown in Table 3.4.

Analysis of the discovered hierarchical parses Fig. 3.12-3.13 present the hierarchical syllabic and phonetic structures that the FullDP system discovers for the two words *globalization* and

collaboration, which are frequently used in the economics lecture. We extract three parse examples for each word from sentences that contain either of them and denote the structures with phone transcriptions for clarity. We now analyze other learning behaviors of our model using these examples.

First, from Fig. 3.12, we can see that the model infers a different sequence of bottom-layer PLUs for each spoken instance of *globalization*. The differences between the three bottom-layer PLU sequences are highlighted by PLUs in red. While the bottom-layer PLUs vary, Fig. 3.12 shows that the noisy-channel model is able to normalize the variations and generate a consistent sequence of top-layer PLUs (106 48 18 31 147 13 6 7 30) for all three instances. Similarly, as shown in Fig. 3.13, distinct bottom-layer PLUs are induced for the three examples of *collaboration*[2]. Once again, the noisy-channel model unifies the differences and produces a unique top-layer PLU representation for the word. This observation demonstrates the effectiveness of the noisy-channel model for capturing phonetic variations. Without the noisy-channel model, the PLU sequences encoding different spoken examples of the same word may vary and are unable to be clustered together.

Note that our model indeed infers a bottom-layer PLU for the consonant /n/ at the end of *globalization* and *collaboration* for each of the instances. However, we find that the acoustic realizations of the consonant /n/ in these instances vary a lot due to the specific contexts within which the two words are spoken. These variations prevents our model from learning a consistent representation for the consonant /n/. As a result, our model acquires the lexical clusters, indicated by $(\cdot)$ in Fig. 3.12 and Fig. 3.13, which correspond to a portion of the word *globalization* and a part of the word *collaboration* respectively. Nevertheless, based on the procedure described in Section 3.5.3 for finding the word label of each discovered lexical unit, we still assign the reference transcriptions *globalization* and *collaboration* to the two discovered lexical clusters. This observation shows that while the noisy-channel model is able to remove many phonetic variations in the speech data, there is still room for improvement. In Section 3.7, we will briefly discuss how we can enhance the quality of the noisy-channel model.

The syllabic structures underlying *globalization* and *collaboration* discovered by the model are indicated by the square brackets in Fig. 3.12 and Fig. 3.13. As discussed earlier, even though the discovered syllabic units do not always match the standard definition of a syllable,

[2]Note that our model does generate identical sequences of bottom-layer PLUs for different spoken tokens of the same word. Whether our model produces the same sequences of bottom-layer PLUs depends on how acoustically similar the spoken tokens are. We utilize the examples in Fig. 3.12 and Fig. 3.13 to illustrate the phonetic variations often observed in the data and to make more insightful analyses on the learning behavior of our model.

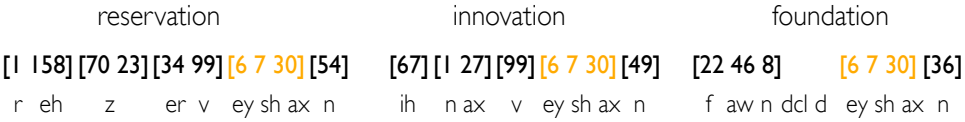


Figure 3.14. More examples of the reuse of the syllabic structure $[6, 7, 30]$.

they often map to structures that are highly reusable. For example, the sound sequence /ey sh ax/ that appears at the end of both *globalization* and *collaboration* is captured by the induced syllabic unit $[6, 7, 30]$ and applied to form both of the words. In addition to *globazliation* and *collaboration*, the syllable structure $[6, 7, 30]$ is also reused to parse many other words in the economics lecture, and we show some of the examples in detail in Fig. 3.14.

Structure reuse is particularly encouraged in our model because of the Pitman-Yor Processes (PYPs) we employ in the adaptor grammar. More specifically, unlike PCFGs, in which the rules used to rewrite nonterminals are selected independently at random from the rule set, the PYPs allow the expansion of a nonterminal symbol to depend on how the symbol has been rewritten in other parses. Therefore, when the adaptor grammar caches the rule Syllable $\rightarrow$ $6, 7, 30$, it increases the probability of choosing this rule to rewrite Syllable in other utterances. Furthermore, given the *rich-gets-richer* clustering property exhibited by the PYPs, the more frequently a rule is used to parse a nonterminal, the more likely it is going to be chosen again. This self-reinforcing behavior of a PYP can be seen from the conditional prior that the PYP assign to a rule r. Take the rule Syllable $\rightarrow 6, 7, 30$ for example. The conditional prior probability of picking this cached rule to expand a nonterminal Syllable is:

$$p(\text{Syllable} \rightarrow 6, 7, 30 | d_{-i}, a^{\text{Syllable}}, b^{\text{Syllable}}) = \frac{C^{+}_{-i}(\text{Syllable} \rightarrow 6, 7, 30) - a^{\text{Syllable}}}{C^{+}_{-i}(\text{Syllable}) + b^{\text{Syllable}}} \quad (3.38)$$

where $C^{+}_{-i}(w)$ is the sum of $C_{-i}(w)$ and the count of w in sentence i, excluding the w associated with the nonterminal being reanalyzed. Eq. 3.38 explicitly expresses the rich-gets-richer phenomenon: the larger $C^{+}_{-i}(r)$ is, the higher prior probability the rule receives.

Moreover, the tendency for structure reuse also stimulates the learning of the noisy-channel model, which is illustrated by the substitution of PLU 30 for PLU 54 shown in the last example of *collaboration* in Fig. 3.13. Given the bottom-layer PLU 54, the preference of using $[6, 7, 30]$ for parsing the word forces the noisy-channel model to use the rule $30 \rightarrow 54$ and infers the top-layer PLU 30. Without this driving force from the higher-level of the model, the learning of the noisy-channel module would be fairly unconstrained, which may prevent the module

Lecture topic	Full50	-AM	-NC	FullDP	-AM	-NC	P&G 2008	Zhang 2013
Economics	12	4	2	12	9	6	11	**14**
Signal processing	16	16	5	**20**	19	14	15	19
Clustering	**18**	17	9	17	**18**	13	16	17
Speaker adaptation	14	14	8	**19**	17	13	13	**19**
Physics	**20**	14	12	**20**	18	16	17	18
Linear algebra	18	16	11	**19**	17	7	17	16

Table 3.5. The number of the 20 target words discovered by each system described in Section 3.5 and by the baseline (P&G, 2008) [146] and by the state-of-the-art system (Zhang, 2013) [190]. The best performance achieved for each lecture is highlighted in bold.

from picking up any useful edit operations. This interaction between the noisy-channel model and the adaptor grammar shows the strength of the proposed joint learning framework.

Quantitative assessments Table 3.5 summarizes the coverage of the top 20 TFIDF scoring words achieved by each of the systems described in Section 3.5 for each lecture. The coverage obtained by the baseline [146] and by the state-of-the-art system [190] are also listed. We highlight the best performance attained for each lecture in bold. From Table 3.5 we can see that the FullDP and Full50 systems consistently outperform the baseline. When compared to the state-of-the-art system, the two full models also perform better for most of the lectures. Note that the difference between the baseline system and the state-of-the-art system is that the latter employs Gaussian posteriorgrams to represent the speech data [191], which have proven to be a more robust speech representation. Although our full systems are only trained on the basic MFCC features, they still discover more lexical units that correspond to the target words for most of the six lectures. We show the full comparison between the coverage over the target words achieved by the FullDP system and by the baseline framework for each lecture in Table 3.6. The red color highlights the words that are found by our model but missed by the baseline, and the blue color denotes the reverse scenario. The black color shows the words that are detected by both systems, while the underlines mark the words that are discovered by neither systems. We can see that the FullDP system consistently finds more target words than the baseline for all the six lectures.

The comparison between the full systems and their -AM counterparts further reveals the effectiveness of the full model. In particular, as shown in Table 3.5, the Full50 system performs

Economics	Signal processing	Clustering	Speaker adaptation	Physics	Linear algebra
flat	frequency	cluster	speaker	electric	matrix
globalization	vocal	distortion	adaptation	zero	row
collaboration	wave	data	model	sphere	zero
india	transform	algorithm	vector	charge	pivot
era	fourier	metric	parameter	plate	equation
flattener	vowel	vector	adapt	symmetry	elimination
dollar	speech	distance	technique	flux	column
china	cavity	speech	utterance	plane	multiply
southwest	signal	split	weight	vector	matrices
argue	tract	assign	likelihood	uniformly	subtract
airline	fold	quantization	estimate	gauss	minus
thousand	sound	dimension	dependent	field	step
outsourcing	acoustic	train	independent	angle	multiplication
really	window	iteration	data	epsilon	exchange
platform	characteristic	plot	recognize	divided	inverse
huge	function	coefficient	speech	vandegraaff	suppose
create	source	mean	error	surface	plus
convergence	velocity	pick	cluster	distribute	negative
connect	tongue	merge	mean	inside	substitution
chapter	noise	criterion	filter	sigma	identity

Table 3.6. The full comparison between the FullDP system and the baseline system for the coverage of the top 20 words with the highest TFIDF scores. The words in black are found by both our model and the baseline. We use underlines to specify words that are learned by neither our model nor the baseline. Finally, the red color denotes words that are discovered by our model but not by the baseline, while the blue color indicates the reverse case.

at least as well as the Full50-AM system for all the lectures; similarly, the FullDP system also surpasses FullDP-AM for almost every lecture. As described in Section 3.5, the -AM systems are identical to the full systems except that they do not contain an acoustic model and thus do not resegment and relabel the speech data during learning. In other words, the advantage the full models have over the -AM systems is that they can refine the bottom-layer PLUs using the information from the higher-level syllabic and lexical structures. This finer performance

Lecture topic	Full50	-AM	FullDP	-AM
Economics	74.4	74.6	74.6	75.0
Signal processing	76.2	76.0	76.0	76.3
Clustering	76.6	76.6	77.0	76.9
Speaker adaptation	76.5	76.9	76.7	76.9
Physics	75.9	**74.9**	75.7	75.8
Linear algebra	75.5	**73.8**	75.5	75.7

Table 3.7. The F1 scores for the phone segmentation task obtained by the full systems and the corresponding -AM systems. Note that since the -AM systems do not resegment the speech data, the F1 scores of the -AM models are the same as those computed by using the segmentations produced by HHMM and DPHMM. The numbers in bold highlight the suboptimal segmentation performance that the initialization system of Full50 achieves compared to that obtained by the initialization system of FullDP.

achieved by the full systems demonstrates the strength of the proposed joint learning framework and exemplifies the synergies of phonetic and lexical learning observed in our model. Finally, by comparing the full systems and the -AM systems to their -NC counterparts, we can see that the noisy-channel model plays a crucial role in the success of inducing lexical units directly from speech data. This observation resonates with Fig. 3.12 and Fig. 3.13. More specifically, the results in Table 3.5 further confirm that without the noisy-channel module, the model would not be able to merge different spoken tokens of the same word into one word cluster.

More evidence on the synergies between phonetic and lexical unit learning Table 3.7 presents the F1 scores attained by the full systems and the -AM systems on the phone segmentation task. Since the -AM systems do not resegment the speech data, the performance of each of the -AM systems is measured by using the segmentation produced by the corresponding initialization systems: the HHMM for Full50-AM and the DPHMM for FullDP-AM.

Let us first take a look at the F1 scores obtained by the -AM systems. From the two -AM columns, we can see that the two initialization systems achieve roughly the same segmentation performance for the **first four** lectures, with the largest performance gap being only 0.4%. Except that they both utilize the boundary elimination method described in Section 2.6 to constrain the hypothesis space for the boundary variables, the two systems are trained independently. Given that the HHMM and the DPHMM are separately trained, this narrow performance gap indicates that the two systems may have already found the optimal segmentation in the hypothesis space. Since our model also looks for the best segmentation in the same hypothesis space, by initializing the boundary variables around the optimum, our model should simply maintain

the segmentation. In particular, as shown in Table 3.7, the full systems also achieve about the same performance as the -AM systems for the first four lectures, with the overall largest performance difference being bounded by 0.4%.

However, what's more interesting is when the initialization system gets stuck at a local optimum. By comparing the performance of the two -AM systems for the **last two** lectures, we can see that the initialization of Full50 converges to local optimums for the two lectures, which are highlighted in bold in Table 3.7. Nonetheless, as shown in Table 3.7, the Full50 system is able to improve the given initial segmentation and reach a similar performance to that accomplished by the FullDP and the initialization of the FullDP systems. This observation indicates that the full model can leverage its knowledge acquired from the higher-level structures in the speech data to refine the segmentation. The superior performance of the Full50 system than that of the Full50-AM system further demonstrates the synergies between the lexical and phonetic unit learning enabled by our model design.

■ 3.7 Chapter Conclusion

In this chapter, we present a probabilistic framework for inferring hierarchical linguistic structures from acoustic signals. Our approach is formulated as an integration of adaptor grammars, a noisy-channel model, and an acoustic model. By encoding words as sequences of syllables and representing syllables as sequences of phonetic units in the parsing grammar, our model is able to jointly discover lexical, syllabic, and phonetic units directly from speech data.

In particular, when tested on lecture recordings, our model demonstrates its capability of discovering frequent subject-specific keywords and acquiring syllabic structures that are highly reusable for constructing different word types. Moreover, by comparing the model to its lesioned counterpart that does not contain a noisy-channel model, we find that modeling phonetic variability plays a critical role in successfully inferring lexical units from speech. A more careful examination on the learning behavior of our model also reveals that it is the proposed joint learning framework that allows the noisy-channel model to capture phonetic variations. Finally, a comparison between the full framework and one without the acoustic model further shows that the strength of the proposed approach comes from the synergies our model stimulates between lexical and phonetic unit acquisition.

The noisy-channel model employed in our framework has demonstrated its ability to normalize phonetic variations. However, the large number of edit operations, $|\mathbb{L}|^2 + |\mathbb{L}| + 1$, associated with each phonetic unit has also hindered efficient computation for the first two in-

ference steps described in Section 3.4. Furthermore, the design of the noisy-channel model ignores potentially useful information, such as the context of a phone unit, which can be leveraged to better capture phonetic variability. A potential future research direction is therefore to develop a noisy-channel model that can make better use of available knowledge to learn phonetic variations while keeping the computation for inference efficient.

We employ the simplest grammar to induce the linguistic structures embedded in speech data. However, there is a wide collection of grammars that can be utilized for this task. For example, by defining a Collocation nonterminal to be a sequence of words, or more specifically, by adding the following rule to our grammar,

$$\text{Sentence} \rightarrow \text{Collocation}^{+}$$
$$\underline{\text{Collocation}} \rightarrow \text{Word}^{+}$$

we can learn the collocation relationships among the discovered lexical units. Furthermore, we can also exploit the grammar shown below to infer the morphological structures in a language from speech data.

$$\text{Sentence} \rightarrow \text{Word}^{+}$$
$$\underline{\text{Word}} \rightarrow \text{Prefix Stem Suffix}$$
$$\underline{\text{Prefix}} \rightarrow \text{PLU}^{+}$$
$$\underline{\text{Stem}} \rightarrow \text{PLU}^{+}$$
$$\underline{\text{Suffix}} \rightarrow \text{PLU}^{+}$$

These grammars can all be easily integrated into our framework for learning rich linguistic structures from speech data. Even though the experimental results presented in this chapter are only preliminary, we believe there is great potential for research in this direction.

Chapter 4

One-shot Learning of Spoken Words

■ 4.1 Chapter Overview

One-shot learning is an ability to acquire and generalize new concepts from one or just a few examples [15, 186]. In this chapter, we propose a computational model for one-shot learning tasks on spoken words and investigate the importance of compositionality in speech for these one-shot learning tasks. To test our hypothesis of the importance of compositionality, we utilize and modify the unsupervised model introduced in Chapter 2 to discover a set of phone-like acoustic units from raw speech data. The automatically inferred acoustic units are then applied to two one-shot learning tasks: *classification* and *generation* of novel spoken words. By comparing our model to humans and baseline systems, which do not exploit any compositional structures in speech, we find that learning acoustic units plays a key role in achieving good performance on the two one-shot learning tasks examined in this chapter.

In Section 4.2, we review some one-shot learning tasks studied in other research fields, as well as two instances of *one-shot* challenges faced by modern Automatic Speech Recognition systems (ASRs). In Section 4.3, we then present an unsupervised model, which is a slight variation of the model introduced in Section 2.4, for discovering the compositional structure in speech. We explain in detail the inference algorithm that is used to train the unsupervised model in Section 4.4. Two one-shot learning tasks are examined in this chapter: one-shot classification, and one-shot generation of spoken words. We describe the experimental setup for the two tasks in Section 4.5, and present the experimental results in Section 4.6. Finally, we conclude this chapter in Section 4.7.

■ 4.2 Related Work

Recently, the concept of one-shot learning has attracted wide interest in various research disciplines, and many computational models have been proposed for one-shot learning challenges

in different contexts [39, 38, 185, 164, 102, 104]. The authors of [104] propose a hierarchical Bayesian model for one-shot classification, and one-shot generation of handwritten characters in the Omniglot corpus [103, 164]. The model exploits the compositionality and the causality embedded in written alphabets across 50 languages. More specifically, the model discovers primitive structures in characters, such as strokes and sub-strokes, along with their spatial relationships, which are shared and re-used across different alphabets. The authors demonstrate that the acquired knowledge in these primitive structures can then be transferred to learn new types of characters, for example, alphabets of a new language. On a task of classifying novel characters based on only one example, the hierarchical Bayesian model is shown to achieve human-level behavior. Furthermore, by generating samples from their model, it is demonstrated that the model can produce handwritten samples that are confusable with those generated by a human in a visual Turing test [30, 29]. In this chapter, we adopt this idea of compositionality, and build an unsupervised model to discover the primitive structures in speech, which are applied to one-shot learning tasks with spoken words.

While the notion of one-shot learning is relatively unexplored for ASR, the problem of out-of-vocabulary (OOV) detection is closely related, which involves detecting new words in speech that are missing from the recognizer's vocabulary. Without the capability of detecting OOV words, speech recognizers may erroneously substitute OOV words with similarly sounding in-vocabulary words. Furthermore, these substitution errors may propagate and affect the recognition performance on words surrounding OOV words. In modern ASR systems, the OOV problem is sometimes addressed by adding sub-word units to the recognizers' vocabulary, allowing new words to be recognized as sequences of sub-word units [6, 155, 156, 145, 10]. Similar to our approach, the OOV detectors represent words as sub-word sequences, which are shared and re-used across spoken words in a language; however, the sub-word units employed in most OOV detectors are induced from a pre-defined phonetic unit set [6, 145]. In contrast, our model induces a set of phone-like units directly from raw speech in an unsupervised manner, resembling how an infant tries to learn the basic sound structures in the speech of his or her native language. Finally, while modern ASR systems must deal with the OOV problem, it is not clear how their performance compares to human performance. Designing experiments that can quantitatively evaluate human ability to detect new vocabulary in continuous speech is a research direction that is worth further exploration.

Another instance of one-shot learning in ASR tasks is query-by-example, whereby systems are given an example of a spoken token and search through a speech corpus for spoken documents that also contain the query word [129, 121, 67, 183, 191, 192, 16, 139, 70]. There are

two prominent research approaches taken to solve the STD problem [122]. The first approach is a more supervised method, in which both the given query example and the spoken documents are transcribed into word or phone sequences by a speech recognizer, on which text-matching techniques can be applied to retrieve documents that contain the keyword [129, 121, 67, 183]. As discussed earlier, although this type of approach exploits the compositionality embedded in speech and represents words as sequences of phones, the sub-word units used in these systems are usually pre-defined, which is different from the unsupervised approach we take. On the other hand, the second main approach to solving the STD problem is based on similarity matching between the spoken query and documents in the feature space, which requires no supervised training [191, 192, 16, 139]. While these approaches are unsupervised and do not rely on any prior knowledge in a language, these methods make no use of the compositional structure in speech. As demonstrated later in this chapter, by exploiting the principle of compositionality, our model can achieve significantly better performance on one-shot classification tasks than baseline systems that ignore the compositional structure in speech.

Finally, there has been computational work on the problem of unsupervised learning of sub-word units in the field of cognitive science [181, 40, 41, 24]. However, most of these models cannot be directly applied to any task that uses raw speech as input data. More clearly, these models usually assume that the phonetic boundaries in speech are known and that the speech data are already converted to a low-dimensional space such as the first and second formant of vowel sounds. In contrast, our model infers sub-word segmentation and sub-word categories from a feature representation that more closely resembles raw speech data.

■ 4.3 Model

In this section, we present our unsupervised model for discovering the compositional structure in speech data. Although often not acknowledged explicitly, the notion of compositionality is ubiquitous in ASR. Take the acoustic model in an ASR system as an example. The acoustic model generally consists of a set of 3-state Hidden Markov Models (HMMs) that represent the phonetic units in a language [87]. These phonetic units can be regarded as the primitive structures of speech in a language, and the HMMs modeling these phonetic units can be recursively concatenated to form larger HMMs to model word pronunciations. Our model builds upon this basic HMM structure to acquire a phone-like representation of a language from speech data.

In Section 2.4, we present a Dirichlet process mixture model with hidden Markov models as the mixtures to discover acoustic units in speech. The DPHMM model discovers acoustic

units by exploiting the *unigram* statistics of the discovered clusters. More specifically, as shown in the first term of Eq. 2.1, to assign a cluster label to a speech segment, the DPHMM model considers how often it has observed samples from each cluster, and utilizes a pseudo count for a new cluster as its prior belief of how likely a segment belongs to each cluster.

However, there is more information than the *unigram* statistics of the acoustic units to be learned and to be exploited for learning. For example, as pointed out in [60], *contextual* information of clusters can help improve the word segmentation performance of unsegmented phone sequences. Therefore, in this section, we present a Bayesian Hierarchical Hidden Markov Model (HHMM) for the task of acoustic unit discovery, which leverages its hierarchical structure to learn not only the *unigram* distribution of the acoustic units, but also the *bigram* transition probabilities between the discovered units.

■ 4.3.1 Bayesian Hierarchical Hidden Markov Model

The hierarchical hidden Markov model consists of two layers: a top layer of states representing the acoustic units to be discovered and a bottom layer of 3-state HMMs, which model the feature dynamics of each acoustic unit. In other words, we use a regular HMM to capture the transition probabilities between the discovered acoustic units (i.e., a phone bigram), and associate each state with a 3-state HMM to model the emission probability of each acoustic unit. An illustration of the proposed HHMM with three acoustic units is shown in Fig. 4.1, in which we use θ_i, for $1 \leq i \leq 3$, to denote an acoustic unit and $\theta_{i,j}$, for $1 \leq j \leq 3$, to denote the 3 sub-states of the HMM associated with the i^{th} acoustic unit.

To formalize the proposed hierarchical hidden Markov model, we review some notation used for the DPHMM model in Section 2.4, and introduce new variables for the hierarchical hidden Markov model. In the model description presented below, we assume there are K states in the top layer of the HHMM. Note that we slightly change the index variables used in Section 2.4 for clarity.

Variable Review

- $x_t \in \mathbb{R}^{39}$: The MFCC features of the training utterances, which are the only observed data in our model. Note that we use x_t to denote the t^{th} speech feature frame in a sentence, and $x_{i,t}$ to denote the t^{th} feature frame in the i^{th} segment of an utterance.

- $b_t \in 0, 1$: The boundary variable associated with each feature vector, which indicates a speech segment boundary between x_t and x_{t+1} if $b_t = 1$, and vice versa.

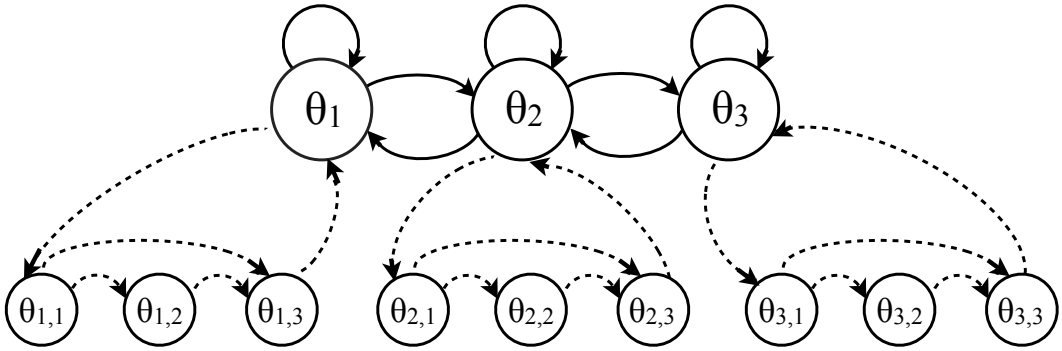


Figure 4.1. An example of the proposed hierarchical hidden Markov model with three discovered acoustic units. Note that we omit the start and the end states of an HMM for simplicity, the top layer HMM is used to model the transition between acoustic units, and the bottom layer HMMs are used to model the feature dynamics of each acoustic unit.

- s_t: The sub-state within a 3-state HMM that x_t is assigned to.
- m_t: The Gaussian mixture component that is used to generate x_t.
- c_i, $1 \leq c_i \leq K$: The cluster label of the i^{th} speech segment in an utterance. Note that in the framework of HHMMs, c_i not only denotes the acoustic unit that a segment is assigned to, but also denotes the id of the state that represents the acoustic unit in the top layer of an HHMM.
- θ_k: The 3-state HMM associated with state k at the top layer of an HHMM.
- θ_0: The prior distribution of θ_k.

Additional Model Variables for HHMM

- $\phi_k \in \mathbb{R}^K$: The transition probability of state k in the top layer of an HHMM, which contains the transition probabilities from acoustic unit k to other acoustic units. We use $\phi_{k,j}$ to indicate the probability of transitioning from the k^{th} to the j^{th} acoustic unit.
- $\beta \in \mathbb{R}^K$: The prior distribution of ϕ_k, which can be thought of as the unigram distribution of the acoustic units. We use β to tie all ϕ_k's together to enforce sharing unigram statistics of the acoustic units for learning bigram transition statistics between the acoustic units. We further impose a symmetric Dirichlet prior distribution, $Dir(\gamma)$, on β.

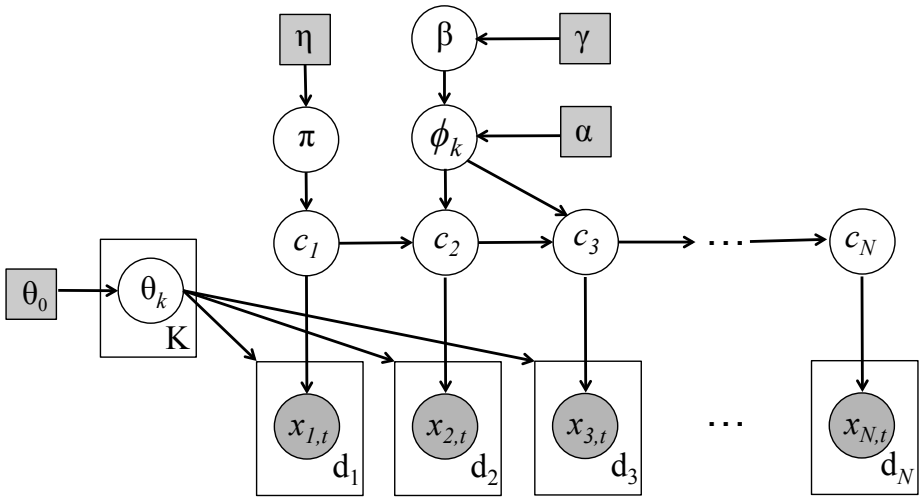


Figure 4.2. The proposed hierarchical hidden Markov model for acoustic unit discovery for N units in an utterance. The shaded circles denote the observed feature vectors, the squares denote the parameters of the priors used in our model, and the unshaded circles are the latent variables of our model.

- $\pi \in \mathbb{R}^K$: The probability distribution of the initial state for the top layer of an HHMM, which is equivalent to the probability of choosing each acoustic unit as the starting unit in an utterance. We use a symmetric Dirichlet distribution, $Dir(\eta)$, as the prior of π.

The graphical representation of the proposed hierarchical hidden Markov model for acoustic unit discovery is shown in Fig. 4.2. In the next section, we describe the generative process implied by the HHMM model.

■ 4.3.2 Generative Process

The generative process for an utterance that has N speech segments can be written as follows. Note that we assume we know the number of segments in an utterance in this example only for clarity. This number is actually not available to our model and needs to be inferred from data. We explain how to infer the number of segments in an utterance in the next section.

To generate an utterance, we first sample the model parameters, as follows.

1. Generate a sample for the initial state probability.

$$\pi \sim Dir(\eta)$$

2. Instantiate the prior distribution β of the transition probabilities.

$$\beta \sim Dir(\gamma)$$

3. Sample the transition probability distribution for each acoustic unit, or each state in the top layer of an HHMM. For $1 \leq k \leq K$,

$$\phi_k \sim Dir(\alpha\beta)$$

4. Parameterize each of the 3-state HMMs that are associated with the acoustic units. For $1 \leq k \leq K$,

$$\theta_k \sim \theta_0$$

The value α in Eq. 3 is the concentration parameter of the distribution from which ϕ_k are drawn, which indicates how similar ϕ_k are to their prior β. To be more clear, the parameter of a Dirichlet distribution can be viewed as pseudo counts of samples from each category before the model observes any data. Therefore, the value of $\alpha\beta$ of the Dirichlet distribution in Eq. 3 can be regarded as our *prior belief* in observing each acoustic cluster following the k^{th} acoustic unit in data. As a result, if we increase the value of α, then we increase the pseudo counts for each category proportional to the prior β, forcing samples of ϕ_k to be more similar to the prior β. On the other hand, if the value of α decreases, then our belief in the prior β also decreases, and thus we rely more on the real observations in the data for generating samples of ϕ_k.

After the model parameters are sampled, we can generate an utterance with N segments as follows.

1. Choose the initial acoustic unit for the utterance.

$$c_1 \sim \pi$$

2. For $2 \leq i \leq N$, select the label of the i^{th} acoustic unit.

$$c_i \sim \phi_{c_{i-1}}$$

3. For $1 \leq i \leq N$, generate the speech features for each segment from θ_{c_i}.

$$x_{i,1}, \ldots, x_{i,d_i} \sim \theta_{c_i}$$

The duration of each segment, d_i, is determined by the number of steps taken to traverse from the beginning to the end of the HMM that the segment is assigned to. Fig. 4.3 illustrates the latent variables in a training example, and how the model parameters π, ϕ_k, and θ_k correspond to the training example.

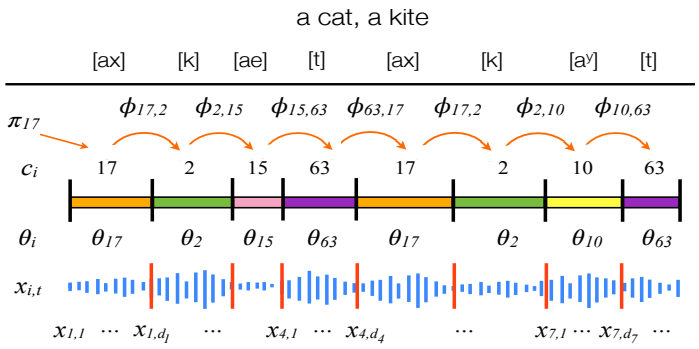


Figure 4.3. An illustration of a typical training example for our model, the latent variables embedded in the utterance, and the model to be learned. Note that only speech data ($x_{i,t}$) is given to the model; the text *a cat, a kite* and the pronunciation are only for illustration. The segment boundaries, indicated by the red bars, the segment cluster labels c_i, as well as the model parameters π, ϕ, and θ all need to be inferred from data.

■ 4.3.3 Comparison to Alternative Models

The HMM-based model presented in [182] is similar to the proposed HHMM model in this section. The major difference is that in the model of [182], the emission probability of each state in the top layer HMM is modeled by a Gaussian Mixture Model (GMM), while in our proposed HHMM, the emission probability of each state is modeled by a 3-state HMM. The design of our model topology allows the states in the top layer HMM to represent units that are more like *phones*, while as shown in [182], the states in the model of [182] tend to correspond to units smaller than a phone. The benefit of learning more phone-like units is that, for example, we can directly substitute expert-defined standard phone sets used in tasks such as speech recognition with the induced acoustic units, and hence reduce the degree of human supervision in those tasks. Even though learning sub-units smaller than phones can potentially achieve the same goals, a mechanism for merging the sub-units into more phone-like units, or new frameworks for completing those tasks based on sub-units, must be developed. These two problems are beyond the scope of this thesis; however, we believe that the two problems are both research directions worth further investigation.

Extension of Hierarchical Hidden Markov Model to a Nonparametric Model

A Bayesian nonparametric extension of the hidden Markov model is Hierarchical Dirichlet Process Hidden Markov Model (HDPHMM) [177, 7], in which a Hierarchical Dirichlet Process (HDP) prior is imposed on the states of the HMM. The HDPHMM has provided a powerful framework for inferring state complexity from data. By applying the same idea to our model, we

can construct the nonparametric counterpart of the HHMM model by imposing an HDP prior on the top layer HMM and having the model automatically infer the number of acoustic units to be discovered for a dataset. While the HDPHMM framework provides a general framework for inferring the data complexity, we decided to use the parametric version of the model for computational efficiency. By setting the number of states in the parametric HHMM to be larger than the expected number of clusters, the finite model can be regarded as a close approximation to the nonparametric model. The inference scheme of setting an upper bound K on the number of clusters that a Dirichlet process finds is referred to as the *degree K weak limit approximation* to the Dirichlet process [72], which has been exploited in the inference algorithms of other nonparametric models and shown to work well [46, 85].

Finally, the model described in [180] for acoustic unit segmentation in speech is also based on an HDPHMM. However, similar to [182], the emission probability of each state is modeled by a GMM. This model structure design is different from that of our model, which utilizes an HMM to capture the temporal dynamics of an acoustic unit in the feature space.

■ 4.4 Inference

In this section, we describe the inference procedure for learning the generative model presented in Section 4.3. The inference procedure consists of three parts: 1) Initialize of the model parameters π, β, ϕ_k, and θ_k; 2) Infer the acoustic unit label c_i for each segment in an utterance; 3) Generate new samples for the model parameters. We construct a Gibbs sampler to learn the model, which is initialized and alternates between the last two steps as follows.

■ 4.4.1 Initialization of the Model Parameters π, β, ϕ_k, and θ_k

To initialize the Gibbs sampler, we obtain samples for the model parameters π, β, ϕ_k and θ_k from their corresponding prior.

$$\pi \sim Dir(\eta) \tag{4.1}$$

$$\beta \sim Dir(\gamma) \tag{4.2}$$

$$\phi_k \sim Dir(\alpha\beta) \quad k = 1, \ldots, K \tag{4.3}$$

$$\theta_k \sim \theta_0 \quad k = 1, \ldots, K$$

■ 4.4.2 Sample Speech Segmentation b_t and Segment Labels c_i

Conditioning on the model parameters, we can construct and generate samples from the posterior distribution of the latent labels of speech segments c_i. However, to construct the conditional posterior distribution, we must integrate out the unknown segmentation within each utterance, which includes the number of segments N, and the location of the segment boundaries. To overcome these two challenges, we employ and modify the message-passing algorithm designed for hidden semi-Markov models [133, 85], and present the modified algorithm for the HHMM model in this section. Based on the message values, we can efficiently compute the posterior probability of each possible segmentation for an utterance as well as the posterior probability of the cluster label of each segment.

Message-passing Algorithm for HHMM

We adopt the notation used in [85] and define B and B^* as follows.

$$B_t(k) \triangleq p(x_{t+1:T}|\mathbb{C}(x_t) = k, b_t = 1) \tag{4.4}$$

$$= \sum_{j=1}^{K} p(\mathbb{C}(x_{t+1}) = j|\mathbb{C}(x_t) = k) B_t^*(j) \tag{4.5}$$

$$= \sum_{j=1}^{K} \phi_{k,j} B_t^*(j) \tag{4.6}$$

$$B_t^*(k) \triangleq p(x_{t+1:T}|\mathbb{C}(x_{t+1}) = k) \tag{4.7}$$

$$= \sum_{d=1}^{T-t} p(x_{t+1:t+d}|\mathbb{C}(x_{t+1}) = k) B_{t+d}(k) \tag{4.8}$$

$$B_T(k) \triangleq 1 \quad k = 1, \ldots, K \tag{4.9}$$

$$B_0(0) = \sum_{k=1}^{K} \pi_k B_1^*(k) \tag{4.10}$$

where we use $x_{t_1:t_2}$ as an abbreviation of $x_{t_1}, \ldots, x_{t_2}$, and T to denote the total number of feature frames in an utterance. The function $\mathbb{C}(x_t)$ maps x_t to the cluster label of the segment

that x_t belongs to. As shown in Eq. 4.4, $B_t(k)$ is defined to be the marginal probability of $x_{t+1:T}$, with all possible segmentations for $x_{t+1:T}$ integrated out, given that x_t is a segment boundary and the speech segment that includes x_t is an observation of the k^{th} acoustic unit. The value of $B^*_t(k)$ contains the marginal probability of $x_{t+1:T}$ given that the segment starting with x_{t+1} has a cluster label k. We compute the value of $B^*_t(k)$ by considering all possible durations of the segment starting with x_{t+1} and multiplying the likelihood of $x_{t+1:t+d}$ being generated by the k^{th} acoustic unit with $B_{t+d}(k)$ that stands for the marginal probability of the features after this segment given $\mathbb{C}(x_{t+d}) = k$. Given the definitions shown in Eq. 4.4 and Eq. 4.7, we can implement this backwards message-passing algorithm using dynamic programming, which allows us to compute the marginal probabilities efficiently. The initialization condition of this backwards message-passing algorithm is specified in Eq. 4.9.

Construct Posterior Distributions of Segmentation and Segment Labels

With the values of B and B^* computed, we can sample the segmentation of an utterance and the cluster labels of the segments recursively starting from the beginning of an utterance. As a quick summary, we use the values stored in $B_t(k)$ to sample the cluster label of the segment starting at $t+1$. Given the cluster label, $j = \mathbb{C}(x_{t+1})$, we exploit the probability carried in $B_t(j)$ to sample the duration d of the segment that starts with x_{t+1}. After this step, the segmentation and cluster labels for feature vectors $x_1 : x_{t+d}$ are determined; we can then recursively use the information carried in $B_{t+d}(j)$ and repeat the procedure to sample the segment label for the next segment. This procedure is repeated until the last segment in the utterance is sampled. We refer to this procedure as the *forwards sampling* step. We take sampling the first segment boundary and segment label as an example and show how to construct the posterior distributions of segment labels and segment duration in detail. The complete forwards sampling algorithm is presented in Alg. 4.4.1.

To construct the posterior probabilities of the cluster label for the segment that starts with x_1, we normalize the K items that contribute to $B_0(0)$ in Eq. 4.10 as follows.

$$p(\mathbb{C}(x_1) = k | x_{1:T}, \pi, \beta, \phi_1, \ldots \phi_k, \theta_1, \ldots \theta_k) = \frac{\pi_k B^*_1(k)}{B_0(0)} \tag{4.11}$$

The cluster label of the first segment c_1 can thus be sampled from the normalized distribution shown in Eq. 4.11. Given the cluster label, the duration of the first segment d can be sampled from the following normalized distribution, which is composed of entries of Eq. 4.8.

Algorithm 4.4.1 Forwards sampling segment boundaries, b_t, and segment labels, c_i

$i \leftarrow 0$
$t \leftarrow 0$
$c_i \leftarrow 0$
while $t < T$ **do**
    $c_{i+1} \leftarrow SampleClusterLabelFromB_t(c_i)$ **% See explanation in Eq. 4.11.**
    $t \leftarrow SampleBoundaryFromB^*_t(c_{i+1})$ **% See explanation in Eq. 4.12.**
    $i \leftarrow i + 1$
end while

$$p(d|c_1, x_{1:T}, \pi, \beta, \phi_1, \dots \phi_k, \theta_1, \dots \theta_k) = \frac{p(x_{1:d}|\mathbb{C}(x_1) = c_1)B_d(c_1)}{B^*_1(c_1)} \tag{4.12}$$

$$= \frac{p(x_{1:d}|\theta_{c_1})B_d(c_1)}{B^*_1(c_1)} \tag{4.13}$$

where $p(x_{1:d}|\theta_{c_1})$ is the likelihood of the speech segment $x_{1:d}$ being generated by the HMM θ_{c_1}, which can be computed using the forward-backward algorithm. At this point, we have found the first segment, which consists of features $x_1, \cdots, x_d$, along with the acoustic unit label c_1. By conditioning on c_1, we can compose the posterior distribution of c_2, the cluster label of the second segment, using Eq. 4.11 with π_k replaced by $\phi_{c_1,k}$, and construct the posterior duration of the second segment as in Eq. 4.12. The sampling procedure alternates between the two steps until a sample for the label of the last segment is generated. Finally, to further improve the efficiency of the computation of B and B^*, we exploit the idea of boundary elimination described in Section 2.5.2 and compute B_t and B^*_t only for t whose associated feature vector x_t is proposed as a potential boundary by the boundary elimination algorithm.

Sample Other Latent Variables Associated with Speech Segments

After sampling the cluster label for each speech segment, we can further generate samples for the state id s_t, and mixture id m_t, for each feature vector in a speech segment. This part of the inference can be carried out with the procedure presented in Section 2.5.

■ 4.4.3 Sample Model Parameters π, β, ϕ_k, and θ_k

Once the samples of c_i are obtained, the conditional posterior distribution of π, ϕ_k, and β can be derived based on the counts of c_i. By conditioning on the state id and the mixture id of each

feature vector, the HMM parameters θ_k can be updated as in Section 2.5. Therefore, in this section, we focus on deriving the posterior distributions of π, β, and ϕ_k.

To update the prior distribution of π, we define $\mathcal{I}_k$ to be the number of times that acoustic unit k appears at the beginning of an utterance in the training data. More formally,

$$\mathcal{I}_k = \sum_{i=1}^{D} \delta(c_1 = k), \tag{4.14}$$

where D is the total number of training utterances in the corpus, and $\delta(\cdot)$ stands for the discrete Kronecker delta. After gathering this count from the data, we can update the prior distribution of π and generate a sample from its posterior distribution.

$$\pi \sim Dir(\eta + \mathcal{I}_1, \eta + \mathcal{I}_2, \ldots, \eta + \mathcal{I}_K) \tag{4.15}$$

Since ϕ_k and β are tied through a hierarchical structure, the derivation of the corresponding posterior distributions is done through a recursive procedure. We define $\mathcal{N}_k$ to be a K-dimensional vector, where the j^{th} entry of $\mathcal{N}_k$ is the number of times the bigram pair $(c_i = k, c_{i+1} = j)$ is observed in the entire corpus based on the sampling results of the previous inference step. More precisely,

$$\mathcal{N}_{k,j} = \sum_{i=1}^{D} \sum_{n=1}^{N_i - 1} \delta(c_n^{(i)}, c_{n+1}^{(i)} = k, j),$$

where N_i is the number of segments in the i^{th} utterance inferred by the Gibbs sampler. In order to update the prior of β, we need to compute the *pseudo* count of each acoustic unit appearing as the second term in all bigram tokens, which can be obtained as follows.

$$\mathcal{M}_j = \sum_{k=1}^{K} \sum_{i=1}^{N_{k,j}} \delta(\nu < \frac{\alpha\beta_j}{i + \alpha\beta_j})$$

where $\frac{\alpha\beta_j}{i+\alpha\beta_j}$ is the probability of generating the acoustic unit j as the second term in a bigram from the prior. For every acoustic unit j that appears as the second term in a bigram, we sample a random variable ν uniformly between 0 and 1 to test whether it is generated from the prior. If it is, then we increase the pseudo count $\mathcal{M}_j$. With the numbers $\mathcal{N}_k$ and $\mathcal{M}_j$ computed, samples for β and ϕ_k can be generated recursively as shown in Eq. 4.16 and Eq. 4.17.

$$\beta \sim Dir(\gamma + \mathcal{M}_1, \ldots, \gamma + \mathcal{M}_K) \tag{4.16}$$

$$\phi_k \sim Dir(\alpha\beta_1 + \mathcal{N}_{k,1}, \ldots, \alpha\beta_K + \mathcal{N}_{k,K}) \quad 1 \leq k \leq K \tag{4.17}$$

■ 4.5 Experimental Setup

Two experiments are designed to validate the hypothesis proposed in this chapter: learning compositional structure is important for one-shot learning of spoken words. The experiments simulate the one-shot learning challenge in two contexts: classification and generation of new spoken words. In this section, we explain the experimental setup for the two tasks in detail as well as describe the training setup for our model and the training corpora used for the experiments.

■ 4.5.1 Corpus of Japanese News Article Sentences

The corpus of Japanese News Article Sentences (JNAS) [73] consists of speech recordings and the corresponding orthographic transcriptions of 153 male and 153 female speakers reading excerpts from the Mainichi Newspaper, and 503 phonetically-balanced (PB) sentences that were chosen by the Interpreting Telephony Research Laboratorie of Advanced Telecommunications Research Institute International in Kyoto. One hundred and fifty five news articles were selected, and each one of the 155 articles was read by one male and one female speaker. Each speaker also read 50 PB sentences. The speech data were recorded by two types of microphones: a Sennheiser HMD410/HMD25-1 or the equivalent head-set microphone, and a desk-top microphone. The data were sampled at a 16 kHz sampling rate, and each sample was quantized into 16 bits. For training the models for the experiments reported in this section, we randomly chose a 10-hour subset of read news articles recorded by using the Sennheiser microphone, which consists of roughly half male and half female speech. There are 150 male talkers and 149 female talkers in this 10-hour subset.

■ 4.5.2 Wall Street Journal Speech Corpus

The Wall Street Journal (WSJ) speech corpus contains read speech of articles drawn from the Wall Street Journal text corpus [51]. An equal number of male and female speakers were chosen to record the corpus for diversity of voice quality and dialect. Two microphones were used for recording: a close-talking Sennheiser HMD414, and a secondary microphone. The speech data

were sampled at 16 kHz and saved as sequences of 16-bit data samples. For training the models, we used a subset of the WSJ corpus, which consists of roughly 12 hours of speech spoken by 26 female and 14 male talkers.

■ 4.5.3 Hyperparameters and Training Details

η	γ	α	K	μ_0	κ_0	α_0	β_0
$\langle 1 \rangle_K$	$\langle 50 \rangle_K$	$\langle 1 \rangle_K$	100	$\boldsymbol{\mu}^d$	5	3	$3/\boldsymbol{\lambda}^d$

Table 4.1. The values of the hyperparameters of the HHMM model, where $\boldsymbol{\mu}^d$ and $\boldsymbol{\lambda}^d$ are the d^{th} entry of the mean and the diagonal of the inverse covariance matrix of training data. We use $\langle a \rangle_K$ to denote a K-dimensional vector, whose entries are all a.

Table 4.1 lists the hyperparameters used for training the HHMM models, in which η, γ and α are the hyperparameters of the prior distributions of π, β, and ϕ_k as shown in Eq. 4.1, Eq. 4.2, and Eq. 4.3. We use $\langle a \rangle_K$ to denote a K-dimensional vector, whose entries are all a. The number of states K, at the top layer of the HHMMs trained for the experiments, is set to 100, which exceeds the size of monophone sets that are usually defined for English and Japanese. For example, 61 unique phones are defined for the TIMIT corpus [52], and these 61 phones are commonly reduced to 48 classes in various experiments reported in the literature [115, 89, 167, 161]. As shown in [142], there are only 16 consonants and 5 vowels defined for the Japanese language. We recall from Section 2.5 that α_0, β_0, μ_0, and κ_0 are components of θ_0, which are the hyperparameters of the prior normal-Gamma distributions for the Gaussian mixtures of each state of the bottom layer 3-state HMMs. We use $\boldsymbol{\mu}^d$ and $\boldsymbol{\lambda}^d$ to denote the d^{th} entry of the mean, and the diagonal of the inverse covariance matrix, of the Gaussian distribution learned from the training data. Finally, we let the Gibbs sampler described in Section 4.4 alternate between the last two inference steps for 10,000 iterations to learn all the models reported in this section.

■ 4.5.4 Classification Task

In the classification task, human subjects and several classifiers, which are built upon the HHMMs, as well as a baseline method, are asked to classify novel spoken Japanese words based on just one example for each of the novel words. More specifically, for each classification trial, 20 Japanese words, which are matched for word length in Japanese characters, are given to a human subject or a classifier. After listening to the 20 Japanese words, the human subject or

1	ダンチョウ	danchyou
2	サンミャク	sanmyaku
3	トランクス	torankusu
4	チョウカン	chyoukan
5	キョクトウ	kyokutou
6	シュッソウ	shyuzou
7	ジョウクウ	jyoukuo
8	ロッキード	rokiido
9	アイジョウ	aijyou
10	エリツィン	erichin
11	ジョウヘキ	jyouheki
12	コガイシャ	kogaishya
13	コウキュウ	koukyuu
14	ヒキノバサ	hikinobase
15	ホリングス	horingusu
16	セレモニー	seremonii
17	ボウケイシ	boukeishi
18	ジョウチョ	jyouchyo
19	ジカンテキ	jikanteki
20	ノゾマシイ	nozomashii

Table 4.2. The stimuli of length 5, along with their approximated pronunciations, used in the classification task for the mismatched gender condition.

the classifier is required to match a *target* word, spoken by a different speaker, to one of the 20 *templates*.

There are two test conditions: 1) matched speaker gender for the target word and the templates and 2) different speaker gender for the target word and the templates. For the matched gender condition, we select 20 words for each word length between 3 and 7 Japanese characters for each gender. Two samples for each of the selected 20 words are extracted from the JNAS corpus and divided into group A and group B. We then create 20 classification trials by using each token in group A as the *target* words and tokens in group B as the *templates*. Likewise, by swapping tokens in group A and group B, we can create another 20 classification trials; therefore, for each gender and each word length, there are 40 classification tests in total. For the

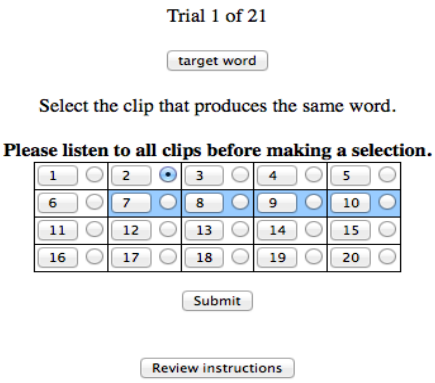


Figure 4.4. A snapshot of the classification trial presented to the participants on Amazon Mechanical Turk. The blue boxes highlight the clips that had not been listened to by the participant.

mismatched gender condition, a similar strategy is applied to create classification trials except that word lengths from 3 to 12 are considered. Table 4.2 shows the 20 selected Japanese words, along with their approximated pronunciations, of length 5 used in the classification trials for the mismatched gender condition.

In the rest of this section, we depict the classification tasks presented to human subjects on Amazon Mechanical Turk (AMT). Also, we explain how the classifiers are built on the HHMM models, as well as describe the baseline system based on Dynamic Time Warping (DTW).

Humans

All participants for the experiments reported in this chapter were recruited via AMT from adults in the USA. Analyses were restricted to native English speakers that do not know any Japanese. Before the classification experiment, participants needed to pass an instruction quiz [22], and there was a practice trial with English words for clarity. Fifty-nine human subjects participated in the classification task in total.

Each of the participants classified new Japanese words in a sequence of displays designed to minimize memory demands. A snapshot of the classification trial presented to the participants on AMT is shown in Fig. 4.4. Participants could listen to the 20 *templates* and the *target* word by clicking the buttons, which they could do more than once to reduce the memory demand. Once the participants found the matched template for the target word, they could click the radio button of the template and submit their response. However, responses were not accepted until all buttons had been tried to prevent the participants from cheating. The clips that had not been listened to before a participant attempted to submit a job would be highlighted by

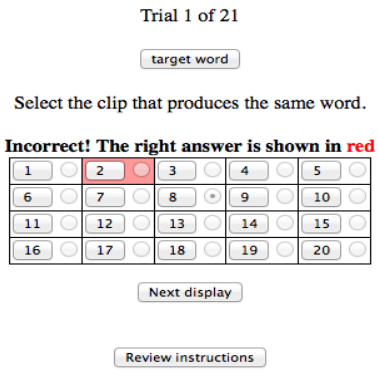


Figure 4.5. A snapshot of the corrective feedback shown to the participants for the classification trial.

blue boxes as demonstrated in Fig. 4.4. Corrective feedback was shown after each response, as shown in Fig. 4.5. Finally, each participant was assigned to one of the two test conditions with matched (5 trials) or different (10 trials) gender. To ensure that learning was indeed one-shot, participants never heard the same template and target words twice and completed only one randomly selected trial for one particular test condition and a specific word length.

Hierarchical Hidden Markov Model Classifiers

Two HHMMs are trained for the classification task. One model is trained on the WSJ subset to simulate an English talker, and the other model is trained on the JNAS subset with all occurrences of the *template* and *target* words excluded. The second model can be viewed as a Japanese speaking child learning words from his/her parents; therefore, we allow the talkers of the *template* and *target* words to overlap the speakers in the 10-hour subset of the JNAS corpus. In fact, for both matched and mismatched gender conditions, all talkers of the template and target words also appear in the 10-hour subset, and each talker contributes to roughly 0.35% of data in the 10-hour data set.

As in the human experiment, for every trial, the model selects one of the 20 template words that best matches the target word. We implement a Bayesian classifier to accomplish the selection process. The Bayesian classifier is defined as follows.

$$\arg\max_{i=1\ldots20} p(X^{(G)}|X^{(i)}) = \arg\max_{i=1\ldots20} \int_{C^{(i)}} p(X^{(G)}|C^{(i)})p(C^{(i)}|X^{(i)})\, dC^{(i)}, \tag{4.18}$$

where $X^{(G)}$ and $X^{(i)}$ are the sequences of Mel-Frequency Cepstral Coefficients (MFCCs) representing the *target* word and each of the *template* words respectively [23]. More specifically,

the spoken *target* and *template* words are each converted to a series of 25 ms 13-dimensional MFCCs and their first- and second-order time derivatives at a 10 ms analysis rate. The variable $C^{(i)}$ denotes all possible unique acoustic unit sequences that underly $X^{(i)}$. The first term in the integral can be interpreted as the likelihood of the sequence of acoustic units contained in $C^{(i)}$ generating the observed feature vectors $X^{(G)}$, and the second term denotes the posterior probability of parsing $X^{(i)}$ into the sequence of acoustic units $C^{(i)}$. Since it is computationally expensive to compute the integral of Eq. 4.18, we approximate the integral with just the $L = 10$ most likely acoustic unit sequences $C^{(i)[1]}, \ldots, C^{(i)[L]}$ that the HHMM model generates for $X^{(i)}$ as in Eq. 4.19.

$$\begin{aligned}
\arg\max_{i=1\ldots20} p(X^{(G)}|X^{(i)}) &= \arg\max_{i=1\ldots20} \int_{C^{(i)}} p(X^{(G)}|C^{(i)}) p(C^{(i)}|X^{(i)})\, dC^{(i)} \\
&\approx \arg\max_{i=1\ldots20} \sum_{l=1}^{L} p(X^{(G)}|C^{(i)[l]}) p(C^{(i)[l]}|X^{(i)}) \\
&\approx \arg\max_{i=1\ldots20} \underbrace{\sum_{l=1}^{L} p(X^{(G)}|C^{(i)[l]}) \frac{p(X^{(i)}|C^{(i)[l]}) p(C^{(i)[l]})}{\sum_{j=1}^{L} p(X^{(i)}|C^{(i)[j]}) p(C^{(i)[j]})}}_{(a)}
\end{aligned} \tag{4.19}$$

It is straightforward to apply the inferred HHMM model parameters π and ϕ_k to compute $p(C^{(i)[l]})$. Assume that $C^{(i)[l]} = c_1^{(i)[l]}, c_2^{(i)[l]}, \ldots, c_N^{(i)[l]}$. The value of $p(C^{(i)[l]})$ can thus be computed as

$$p(C^{(i)[l]}) = \pi_{c_1^{(i)[l]}} \prod_{j=1}^{N-1} \phi_{c_j^{(i)[l]}, c_{j+1}^{(i)[l]}}. \tag{4.20}$$

To compute $p(X^{(i)}|C^{(i)[l]})$, we concatenate the 3-state HMMs associated with the acoustic units in $C^{(i)[l]}$ to form a large HMM and use the forward-backward algorithm to obtain the likelihood of observing $X^{(i)}$ given $C^{(i)[l]}$. Finally, as in [104], we find marginally better performance by using Eq. 4.21 instead of Eq. 4.18,

$$\arg\max_{i=1\ldots20} p(X^{(G)}|X^{(i)}) = \arg\max_{i=1\ldots20} p(X^{(G)}|X^{(i)}) \frac{p(X^{(i)}|X^{(G)})}{p(X^{(i)})}. \tag{4.21}$$

Both sides of Eq. 4.21 are equivalent if inference is exact, but due to the approximations, we include $\frac{p(X^{(i)}|X^{(G)})}{p(X^{(i)})}$ to regularize the classifier. The value of $p(X^{(i)}|X^{(G)})$ can be computed by using Eq. 4.19-(a) with the roles of $X^{(i)}$ and $X^{(G)}$ swapped, and the value of $p(X^{(i)})$ is approximated as follows.

Algorithm 4.5.1 Dynamic Time Warping for Two Speech Feature Sequences $X^{(1)}$ and $X^{(2)}$

% Let $X^{(1)} = x_1^{(1)}, \ldots, x_{N_1}^{(1)}$
% Let $X^{(2)} = x_1^{(2)}, \ldots, x_{N_2}^{(2)}$
$D_{min} := array[N_1 + 1][N_2 + 1]$ **% Initialize a 2-dimensional array**
$D_{min}[0][0] = 0$
for $i = 1, \ldots, N_1$ **do**
    $D_{min}[i][0] = \infty$
end for
for $i = 1, \ldots, N_2$ **do**
    $D_{min}[0][i] = \infty$
end for
for $i = 1, \ldots, N_1$ **do**
    for $j = 1, \ldots, N_2$ **do**
        $d = \cos(x_i^{(1)}, x_j^{(2)})$ **% Cosine distance between** $x_i^{(1)}$ **and** $x_j^{(2)}$
        $D_{min}[i][j] = d + \min(D_{min}[i-1][j], D_{min}[i][j-1], D_{min}[i-1][j-1])$
    end for
end for
return $D_{min}(N_1, N_2)$

$$
\begin{aligned}
p(X^{(i)}) &\sim \sum_{l=1}^{L} p(X^{(i)}, C^{(i)[l]}) \\
&= \sum_{l=1}^{L} p(X^{(i)} | C^{(i)[l]}) p(C^{(i)[l]})
\end{aligned}
$$

Dynamic Time Warping

We compare the HHMM classifier against a DTW-based classifier, which is the baseline for the classification experiment. DTW is a classic algorithm that is widely used to measure similarity between two sequences of speech features [163, 162, 138, 135, 168]. It requires no learning, and the sequences are compared by computing an optimal non-linear warping path and measuring the distance between the aligned sequences given the optimal warping path. More specifically, for our classification task, the DTW-based classifier selects the best matching template with the target word by using the following rule.

$$i^* = \arg\max_{i=1...20} \mathbb{D}(X^{(G)}, X^{(i)}) \tag{4.22}$$

where $\mathbb{D}(X^{(G)}, X^{(i)})$ denotes the cost of the optimal warping path between the features of $X^{(G)}$ and $X^{(i)}$, which can be computed using the dynamic programming algorithm shown in Alg. 4.5.1.

■ 4.5.5 Generation Task

Humans generalize in many other ways beyond classification. The other one-shot learning task we investigate in this chapter is *generation*. Can English speakers generate compelling new examples of Japanese words after hearing just one example of a Japanese word? Can the models do the same? Here we describe the experimental design for testing human subjects and several models on the task of one-shot spoken word generation.

In each generation trial, the human subjects and the HHMM models listen to a male Japanese word example. The human participants and the models are then asked to generate or synthesize the given word. The male Japanese word examples are the same as those used in the classification experiment for the different gender condition. The ultimate evaluation is to run a *Turing test* on these generated or synthesized words [30, 29], in which other human subjects, who are referred to as human judges in the rest of the section, are asked to distinguish between human-generated and model-synthesized speech tokens. However, since synthesizing natural speech is still a difficult technical challenge [189], and since the HHMM models are not tailored for the purpose of speech synthesis, we develop another evaluation method for the generation task. As a result, performance is measured by asking human judges to classify the generated and the synthesized examples into the intended class as in the classification task. The classification accuracy rate can then be an indicator of exemplar quality.

Humans

A snapshot of the generation task displayed to the participants on AMT is shown in Fig. 4.6. Ten participants were recruited for this task. Each participant was assigned a different word length (3 to 12), and then must complete twenty trials of recording using a computer microphone. Participants were allowed to re-record until they were satisfied with the quality. This procedure collected one sample per stimulus used in the classification task for the mismatched gender condition. Therefore, there were two hundred spoken samples collected from the human subjects in total.

Figure 4.6. A snapshot of the generation trial displayed to the participants on Amazon Mechanical Turk.

Hierarchical Hidden Markov Models for Speech Synthesis

The two HHMM models that are used to build the Bayesian classifiers in the classification experiment are also utilized to synthesize speech for the generation task. We refer to these two models as the Japanese HHMM and the English HHMM, as they are trained on the JNAS and WSJ corpora respectively. For the generation task, both models listen to the same new Japanese words as the human subjects, and then synthesize new examples. To synthesize speech, the models first parse each given Japanese word example, denoted as $X^{(i)}$, into a sequence of acoustic units $\hat{C}^{(i)}$ as follows.

$$\begin{aligned} \hat{C}^{(i)} &= \arg\max_{C^{(i)}} p(C^{(i)}|X^{(i)}) \\ &= \arg\max_{C^{(i)}} p(X^{(i)}|C^{(i)})p(C^{(i)}) \end{aligned} \tag{4.23}$$

where the first term $p(X^{(i)}|C^{(i)})$ in Eq. 4.23 can be computed by using the forward-backward algorithm, and the second term can be computed as in Eq. 4.20. The entire decoding task for finding the most likely $\hat{C}^{(i)}$ can also be solved via the forward-backward algorithm. With $\hat{C}^{(i)}$ decoded, we can then concatenate the 3-state HMMs associated with each acoustic unit in $\hat{C}^{(i)}$ to form a whole-word HMM for the Japanese word sample $X^{(i)}$ and forward generate MFCC features from this whole-word HMM.

While it is easy to forward sample MFCC features from the whole-word HMM, we adopt the procedure used by most HMM-based speech synthesis systems [178, 189], and generate the mean vector of the most weighted Gaussian mixture component for each sub-state in the whole-word HMM. Furthermore, HMM-based synthesis systems usually have a duration model for explicitly modeling the length of each acoustic unit [187]. Since this information is missing

from our HHMM model, we force the synthesized speech to have the same duration as the given example. In particular, assume that $\hat{C}^{(i)} = c_1^{(i)}, \ldots, c_N^{(i)}$. For each $c_j^{(i)}$, for $1 \leq j \leq N$, we count the number of frames in $X^{(i)}$ that are mapped to $c_j^{(i)}$ and generate samples from $\theta_{c_j^{(i)}}$ evenly from the 3 sub-states of $\theta_{c_j^{(i)}}$. Finally, to improve the quality of the speech, we extract the fundamental frequency information from the given spoken word example by using [1]. After the information of the fundamental frequency is combined with the generated MFCCs, the features are inverted to audio by the speech processing tool provided in [32].

Last but not least, to more directly study the role of the learned units, we include several lesioned HHMMs, referred to as the *one-unit* model and the *unit-replacement models*. The *one-unit* model is trained on the JNAS corpus with only one acoustic unit allowed to be learned during training, providing the model with a rather limited notion of compositionality. Finally, several unit-replacement models are adapted from the Japanese HHMM, English HHMM, and the one-unit model. To synthesize speech, the unit-replacement models take the inferred unit sequence $\hat{C}^{(i)}$ and perturb the units by randomly replacing a subset with other units. After the first unit is replaced, additional units are also replaced until a 25% or 50% *noise level* is exceeded, as measured by the fraction of replaced feature frames in $X^{(i)}$.

Evaluation Procedure

To evaluate the quality of the human-generated and the machine-synthesized speech tokens, 30 participants are recruited on AMT to classify a mix of speech clips produced by both the human subjects and the HHMM models. The trials appear in the same way as in the classification task, where the top button plays a human-generated or a machine-synthesized *target* word. The other 20 buttons play the original Japanese recordings, matched for word length within a trial as in the classification experiments. Since the synthesized examples are based on male clips, only the female clips are used as the 20 templates. There is one practice trial (in English) followed by 50 trials with the target example drawn uniformly from the set of all generated or synthesized samples across conditions. Since the example sounds vary in quality and some are hardly speech-like, participants are warned that the sound quality varies, may be very poor, or may sound machine generated. Also, the instructions and practice trial are changed from the classification task to include a degraded rather than a clear target word clip. All clips are normalized for volume.

4.6 Results and Analysis

4.6.1 Classification

Unit (%)	Matched gender	Mismatched gender
Humans	2.6	2.9
Japanese HHMM	7.5	21.8
English HHMM	16.8	34.5
DTW	19.8	43.0

Table 4.3. One-shot classification error rates. The table shows the means of the classification error rates across different word lengths for each test condition (matched or mismatched gender) and for each learner (human subjects, HHMM classifiers, or DTW classifier).

The one-shot classification error rates made by the human subjects, HHMM Bayesian classifiers, and the DTW classifier are shown in Table 4.3. The table shows the means of the classification error rates across different word lengths within each test condition. Human subjects made fewer than 3% errors for both matched gender and mismatched gender test conditions, which is the best performance achieved by all learners (human subjects, the HHMM classifiers, and the DTW classifier).

While the best performing model still generates a 5% higher classification error rate than human subjects, both the Japanese HHMM and the English HHMM beat the DTW baseline in the two test conditions. Particularly, the English HHMM is trained on a corpus that mismatches the target and template Japanese speech used in the task in language, speakers and recording conditions. Even though the English HHMM needs to deal with all these corpus mismatch problems that the DTW baseline does not encounter mostly, the English HHMM still outperforms the DTW baseline. The better performance achieved by the English HHMM in a tougher learning condition demonstrates that discovering the basic structures in speech helps the model generalize to learn new words in one-shot classification tasks. Furthermore, the performance obtained by the English HHMM also suggests that the acoustic units inferred from an English corpus can be utilized to learn new words in Japanese. This observation shows that knowledge about basic acoustic units learned for one language can be transferred to learn another language, especially for the language pair of English and Japanese, where the phone set of the former is roughly a super set of the latter [142, 141].

The gap between human and machine performance is much larger for the HHMM trained on English than the model trained on Japanese, which could be a product of many factors

including mismatches in the WSJ corpus and the JNAS corpus as discussed earlier. While the English-trained model may be more representative of the human participants, the Japanese-trained model is more representative of everyday word learning scenarios, like a child learning words spoken by a familiar figure.

The performance of all models degrades on the mismatched gender test condition, which is not fully unexpected given the simple MFCC feature representation that is used for training. The inferior performance for the mismatched gender condition indicates that the HHMMs might have learned different sets of acoustic units for different genders, which could be mitigated by using more robust feature representations for training the HHMMs. Even though there is still a small performance gap between the best HHMM and the human subjects, the consistent superior performance of the HHMMs compared to DTW evidently supports the hypothesis that compositionality is an important facilitator of one-shot learning.

■ 4.6.2 Generation

Several participants commented that the classification task on the generated speech samples was too long or too difficult. Participants spent from 19 to 87 minutes on the task, and there was correlation between accuracy and time ($R=0.58$, $p<0.001$). In a conservative attempt to eliminate guessing, two participants were removed for listening to the *target word* fewer than twice on average per trial (6 times was the experiment average). Note that this removal made little difference in the pattern of results, which is shown in Fig. 4.7. The x-axis of Fig. 4.7 indicates the fraction of the feature vectors that are replaced by the *unit-replacement* model during the synthesis process, and the y-axis shows the classification accuracy rate by using the generated or synthesized samples as the target words in the classification task. A higher classification accuracy rate indicates a better quality of the generated or synthesized speech tokens.

Overall, English speakers achieve an average classification accuracy rate of 76.8%. The best HHMM is trained on the JNAS corpus and achieves an accuracy rate of 57.6%. The *one-unit* model sets the baseline at 17%, and performance of both the Japanese HHMM and the English HHMM decrease towards this baseline as more units are randomly replaced. As with the classification experiment, the Japanese HHMM is superior to the English HHMM and synthesizes Japanese words that are more similar to the given Japanese words in the one-shot generation task.

The high performance achieved by the human subjects suggests that even naive learners can generate compelling new examples of a foreign word successfully, at least in the case

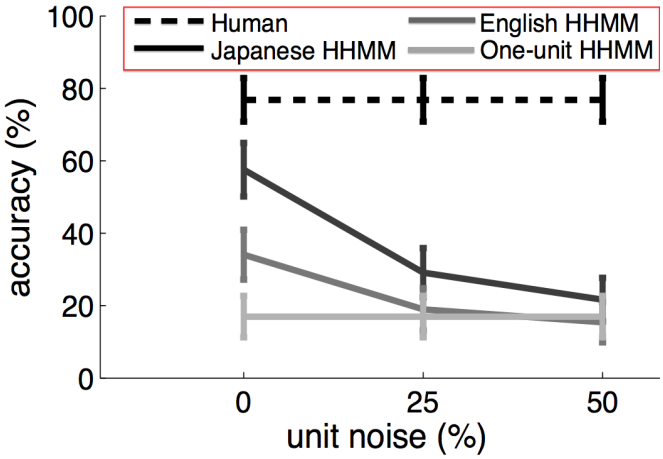


Figure 4.7. Percent of synthesized examples that human judges classified into the correct spoken word category. Error bars are 95% binomial proportion confidence intervals based on the normal approximation.

of Japanese where the Japanese phone set is approximately a subset of the English phone set [142, 141]. The two full HHMMs do not perform as well as humans do. However, given the fact that the one-unit and unit-replacement models only differ from the full HHMMs by their impoverished unit structure, the better results achieved by the full HHMM models still highlight the importance of unit learning in the one-shot generation task.

Replication

A number of participants commented on the task difficulty. Since human and machine voices were intermixed as the target words in the classification task, it is possible that some participants just gave up on trying to interpret any of the machine generated speech. We investigate this possibility by running another batch of experiments, in which each trial consists of speech tokens generated or synthesized by only one system. More specifically, forty-five participants were recruited and assigned to one of three conditions: speech generated by humans, by the Japanese HHMM, or by the English HHMM. Three participants were removed for knowing some Japanese, and three more were removed by the earlier guessing criterion. The results largely confirm the previous numbers. The human-generated speech scores an accuracy rate of 80.8% on average; also, the Japanese HHMM and the English HHMM score 60% and 27.3%. The previous numbers are 76.8%, 57.6%, and 34.1%, respectively.

■ 4.7 Chapter Conclusion

Although the topic of one-shot learning has attracted much interest in research fields such as cognitive science and computer vision, there have not yet been many computational models that have been developed for one-shot learning tasks for speech. In this chapter, we presented a Bayesian Hierarchical Hidden Markov Model (HHMM) that infers the compositional structure in speech of a language for one-shot learning tasks. We compare several learners that are built on HHMMs with human subjects on one-shot classification and one-shot generation of new Japanese words.

The experimental results show that humans are very accurate classifiers, and they can produce compelling examples of Japanese words, even with no experience speaking Japanese. The best performing HHMM is trained on the corpus of Japanese News Article Sentences, and it comes within 5% of human performance on the task of one-shot classification. While facing many challenges resulting from corpus mismatch, the HHMM trained on the Wall Street Journal corpus still outperforms the DTW baseline, which does not take advantage of the compositional structure in speech, for the one-shot classification task. The better performance achieved by the English HHMM compared to the DTW baseline also demonstrates that the phonological knowledge the model acquires from an English data set can be transferred across languages to learn new Japanese words. Furthermore, while the HHMMs do not produce Japanese samples that are as accurate as those generated by human subjects, the one-unit model and the unit-replacement model that are designed to dismiss the compositional structure in speech all deteriorate the quality of the synthesized Japanese speech, which further supports the importance of compositionality for one-shot learning as suggested in previous work [104, 102].

Chapter 5

Joint Learning of Acoustic Units and Word Pronunciations

5.1 Chapter Overview

For creating an Automatic Speech Recognition (ASR) system for a new language, the usual requirements are: first, a large speech corpus with word-level annotations; second, a pronunciation dictionary that essentially defines a phonetic inventory for the language as well as word-level pronunciations, and third, optional additional text data that can be used to train the language model. Given these data and some decision about the signal representation, e.g., Mel-Frequency Cepstral Coefficients (MFCCs) [23] with various derivatives, as well as the nature of the acoustic and language models such as 3-state HMMs, and n-grams, iterative training methods can be used to effectively learn the model parameters for the acoustic and language models. Although the details of the components have changed through the years, this basic ASR formulation was well established by the late 1980's, and has not really changed much.

One of the interesting aspects of this formulation is the inherent dependence on the dictionary, which defines both the phonetic inventory of a language, and the pronunciations of all the words in the vocabulary. The dictionary is arguably the cornerstone of a speech recognizer as it provides the essential transduction from sounds to words. Unfortunately, the dependency on this resource is a significant impediment to the development of speech recognizers for new languages, since the creation of a pronunciation lexicon requires a lot of expert knowledge and remains a highly inefficient process.

The existence of an expert-defined dictionary in the midst of stochastic speech recognition models is somewhat ironic, and it is worth asking why it continues to receive special status after all these years. Why can we not learn the inventory of sounds of a language and associated word pronunciations automatically from data, much as we learn the acoustic model parameters? Can

we apply the acoustic unit discovery model presented in Chapter 2 to find word pronunciations, and eventually develop a fully data-driven training paradigm for ASR systems?

In this chapter, we propose an unsupervised alternative – requiring no language-specific knowledge – to the conventional manual approach for creating pronunciation dictionaries. We present a hierarchical Bayesian model, which jointly discovers the phonetic inventory and the Letter-to-Sound (L2S) mapping rules in a language using only transcribed data. When tested on a corpus of spontaneous queries, the results demonstrate the superiority of the proposed joint learning scheme over its sequential counterpart, in which the latent phonetic inventory and L2S mappings are learned separately. Furthermore, the recognizers built with the automatically induced lexicon consistently outperform grapheme-based recognizers, and even approach the performance of recognition systems trained using conventional supervised procedures.

We organize the rest of this chapter as follows. In Section 5.2, previous work in learning word pronunciations and training ASR systems without lexicons is briefly reviewed. In Section 5.3, we formulate the problem, present the proposed model, and define the generative process implied by our approach. The inference algorithm is discussed in detail in Section 5.4. We describe the experimental setup and the procedure for building a speech recognizer out of our model in Section 5.5. The experimental results and the induced lexicon are analyzed in Section 5.6. Finally, we draw conclusions in Section 5.7.

■ 5.2 Related Work

Various algorithms for learning sub-word based pronunciations were proposed in [113, 49, 4, 144]. In these approaches, spoken samples of a word are gathered, and usually only one single pronunciation for the word is derived based on the acoustic evidence observed in the spoken samples. The major difference between our work and these previous works is that our model learns word pronunciations in the context of letter sequences. Specifically, our model learns letter pronunciations first and then concatenates the pronunciation of each letter in a word to form the word pronunciation. The advantage of our approach is that pronunciation knowledge learned for a particular letter in some arbitrary word can subsequently be used to help learn the letter's pronunciation in other words. This property allows our model to potentially learn better pronunciations for less frequent words.

Our work is closely related to automatically deriving phonetic units, which first received attention in the late 1980's [56, 113]. However, it did not receive significant attention again until the mid to late 2000's, when the authors of [50, 169] began investigating the use of self-

organizing units for keyword spotting and other tasks for languages with limited linguistic resources. Others who have explored the unsupervised space include [182, 74, 111]. More recently, the author of [58] defined a series of increasingly unsupervised learning challenges for speech processing that progressively reduce the amount of linguistic resources available for training. The work reported in this chapter represents the first step away from normal supervised training, where the phonetic units and pronunciation dictionary are not available, but parallel annotated speech data is. The framework proposed in this chapter is deployed based on the DPHMM introduced in Chapter 2. Particularly, in addition to just learning a set of phonetic units as the DPHMM, the model introduced in this chapter further learns the connection between the written form and the spoken form of a language.

Finally, the concept of creating a speech recognizer for a language with only orthographically annotated speech data has also been explored previously by means of graphemes. This approach has been shown to be effective for alphabetic languages with relatively straightforward grapheme to phoneme transformations and does not require any unsupervised learning of units or pronunciations [91, 173]. As we explain in later sections, grapheme-based systems can actually be regarded as a special case of our model; therefore, we expect our model to have greater flexibilities for capturing pronunciation rules of graphemes.

■ 5.3 Model

The goal of our model is to induce a word pronunciation lexicon from spoken utterances and their corresponding word transcriptions. No other language-specific knowledge is assumed to be available, including the phonetic inventory of the language. To achieve the goal, our model needs to solve the following two tasks:

- Discover the phonetic inventory.
- Reveal the latent mapping between the letters and the discovered phonetic units.

We propose a hierarchical Bayesian model for jointly discovering the two latent structures from an annotated speech corpus. Before presenting our model, we first describe the key latent and observed variables of the problem.

Letter (l_i^m) We use l_i^m to denote the i^{th} letter observed in the word transcription of the m^{th} training sample. To be sure, a training sample involves a speech utterance and its corresponding text transcription. The letter sequence composed of l_i^m and its context, namely

$l^m_{i-\kappa}, \cdots, l^m_{i-1}, l^m_i, l^m_{i+1}, \cdots, l^m_{i+\kappa}$, is denoted as $\vec{l}^m_{i,\kappa}$. Although l^m_i is referred to as a *letter* in this paper, it can represent any *character* observed in the text data, including space and symbols indicating sentence boundaries. The set of unique characters observed in the data set is denoted as G. For notation simplicity, we use $\mathcal{L}_\kappa$ to denote the set of letter sequences of length $2\kappa+1$ that appear in the dataset and use $\vec{l}_\kappa$ to denote the elements in $\mathcal{L}_\kappa$. Finally, $\mathbb{P}(\vec{l}_\kappa)$ is used to represent the *parent* of $\vec{l}_\kappa$, which is a substring of $\vec{l}_\kappa$ with the first and the last characters truncated.

Number of Mapped Acoustic Units (n^m_i) Each letter l^m_i in the transcriptions is assumed to be mapped to a certain number of phonetic units. For example, the letter *x* in the word *fox* is mapped to 2 phonetic units /k/ and /s/, while the letter *e* in the word *lake* is mapped to 0 phonetic units. We denote this number as n^m_i, and limit its value to be 0, 1 or 2 in our model. The value of n^m_i is always unobserved and needs to be inferred by the model.

Identity of the Acoustic Unit ($c^m_{i,p}$) For each phonetic unit that l^m_i maps to, we use $c^m_{i,p}$, for $1 \leq p \leq n^m_i$, to denote the identity of the phonetic unit. Note that the phonetic inventory that describes the data set is unknown to our model, and the identities of the phonetic units are associated with the acoustic units discovered automatically by our model.

Speech Feature x^m_t The observed speech data in our problem are converted to a series of 25 ms 13-dimensional MFCCs [23] and their first- and second-order time derivatives at a 10 ms analysis rate. We use $x^m_t \in \mathbb{R}^{39}$ to denote the t^{th} feature frame of the m^{th} utterance.

■ 5.3.1 Generative Process

We present the generative process for a single training sample (i.e., a speech utterance and its corresponding text transcription); to keep notation simple, we discard the index variable m in this section.

For each l_i in the transcription, the model generates n_i, given $\vec{l}_{i,\kappa}$, from the 3-dimensional categorical distribution $\phi_{\vec{l}_{i,\kappa}}(n_i)$. Note that for every unique $\vec{l}_{i,\kappa}$ letter sequence, there is an associated $\phi_{\vec{l}_{i,\kappa}}(n_i)$ distribution, which captures the fact that the number of phonetic units a letter maps to may depend on its context. In our model, we impose a Dirichlet distribution prior $Dir(\eta)$ on $\phi_{\vec{l}_{i,\kappa}}(n_i)$.

If $n_i = 0$, l_i is not mapped to any acoustic units and the generative process stops for l_i; otherwise, for $1 \leq p \leq n_i$, the model generates $c_{i,p}$ from:

$$c_{i,p} \sim \pi_{\vec{l}_{i,\kappa},n_i,p} \tag{5.1}$$

where $\pi_{\vec{l}_{i,\kappa},n_i,p}$ is a K-dimensional categorical distribution, whose outcomes correspond to the phonetic units discovered by the model from the given speech data. Eq. 5.1 shows that for each combination of $\vec{l}_{i,\kappa}$, n_i and p, there is an unique categorical distribution. An important property of these categorical distributions is that they are coupled together such that their outcomes point to a consistent set of phonetic units. In order to enforce the coupling, we construct $\pi_{\vec{l}_{i,\kappa},n_i,p}$ through a hierarchical process.

$$\beta \sim Dir(\gamma) \tag{5.2}$$

$$\pi_{\vec{l}_{i,\kappa},n_i,p} \sim Dir(\alpha_\kappa \beta) \text{ for } \kappa = 0 \tag{5.3}$$

$$\pi_{\vec{l}_{i,\kappa},n_i,p} \sim Dir(\alpha_\kappa \pi_{\vec{l}_{i,\kappa-1},n_i,p}) \text{ for } \kappa \geq 1 \tag{5.4}$$

To interpret Eq. 5.2 to Eq. 5.4, we envision that the observed speech data are generated by a K-component mixture model, of which the components correspond to the phonetic units in the language. As a result, β in Eq. 5.2 can be viewed as the mixture weight over the components, which indicates how likely we are to observe each acoustic unit in the data overall. By adopting this point of view, we can also regard the mapping between l_i and the phonetic units as a mixture model, where $\pi_{l_i,n_i,p}$[1] represents how probable l_i is mapped to each phonetic unit, given n_i and p. We apply a Dirichlet distribution prior parametrized by $\alpha_0\beta$ to $\pi_{l_i,n_i,p}$, as shown in Eq. 5.3. With this parameterization, the mean of $\pi_{l_i,n_i,p}$ is the global mixture weight β, and α_0 controls how similar $\pi_{l_i,n_i,p}$ is to the mean. More specifically, for large $\alpha_0 \gg K$, the Dirichlet distribution is highly peaked around the mean; in contrast, for $\alpha_0 \ll K$, the mean lies in a valley. The parameters of a Dirichlet distribution can also be viewed as pseudo-counts for each category. Eq. 5.4 shows that the prior for $\pi_{\vec{l}_{i,\kappa},n_i,p}$ is seeded by pseudo-counts that are proportional to the mapping weights over the phonetic units of l_i in a shorter context. In other words, the mapping distribution of l_i in a shorter context can be thought of as a back-off distribution of l_i's mapping weights in a longer context.

Each component of the K-dimensional mixture model is linked to a 3-state Hidden Markov Model (HMM). These K HMMs are used to model the phonetic units in the language [76]. The emission probability of each HMM state is modeled by a diagonal Gaussian Mixture Model (GMM). We use θ_c to represent the set of parameters that define the c^{th} HMM, which includes the state transition probability and the GMM parameters of each state emission distribution. The conjugate prior of θ_c is denoted as $H(\theta_0)$. More specifically, $H(\theta_0)$ includes a Dirichlet prior for the transition probability of each state, and a Dirichlet prior for each mixture weight

[1] An abbreviation of $\pi_{\vec{l}_{i,0},n_i,p}$

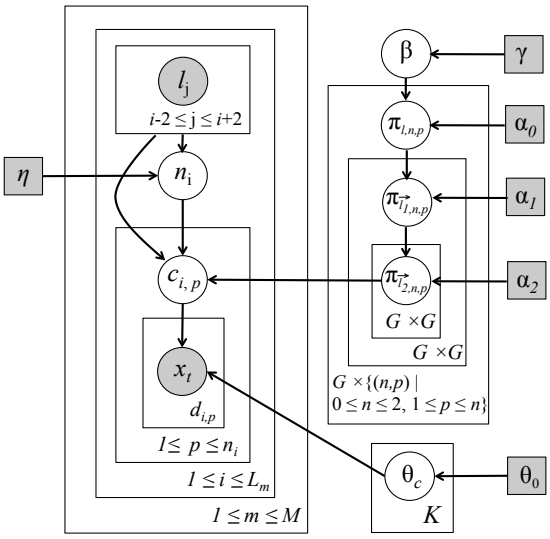


Figure 5.1. The graphical representation of the proposed hierarchical Bayesian model. The shaded circle denotes the observed text and speech data, and the squares denote the hyperparameters of the priors in our model. See Section 5.3 for a detailed explanation of the generative process of our model.

of the three GMMs, and a normal-Gamma distribution for the mean and precision of each Gaussian mixture in the 3-state HMM.

Finally, to finish the generative process, for each $c_{i,p}$ we use the corresponding HMM $\theta_{c_{i,p}}$ to generate the observed speech data x_t, and the generative process of the HMM determines the duration, $d_{i,p}$, of the speech segment. The complete generative model, with κ set to 2, is depicted in Fig. 5.1; M is the total number of transcribed utterances in the corpus, and L_m is the number of letters in utterance m. The shaded circles denote the observed data, and the squares denote the hyperparameters of the priors used in our model. Lastly, the unshaded circles denote the latent variables of our model, for which we derive inference algorithms in the next section.

■ 5.4 Inference

We employ Gibbs sampling [55] to approximate the posterior distribution of the latent variables in our model. In the following sections, we first present a message-passing algorithm for block-sampling n_i and $c_{i,p}$, and then describe how we leverage acoustic cues to accelerate the computation of the message-passing algorithm. Note that the block-sampling algorithm for n_i and $c_{i,p}$ can be parallelized across utterances. Finally, we briefly describe how we use a voice activity detector to train an initial silence model, and discuss the inference procedures for $\phi_{\vec{l}_\kappa}$,

$\pi_{\vec{l}_\kappa,n,p}$, β, θ_c.

■ 5.4.1 Block-sampling n_i and $c_{i,p}$

To understand the message-passing algorithm in this study, it is helpful to think of our model as a simplified Hidden Semi-Markov Model (HSMM), in which the letters represent the states and the speech features are the observations. However, unlike in a regular HSMM, where the state sequence is hidden, in our case, the state sequence is fixed to be the given letter sequence. With this point of view, we can modify the message-passing algorithms of [132] and [85] to compute the posterior information required for block-sampling n_i and $c_{i,p}$.

Let $\mathbb{L}(x_t)$ be a function that returns the index of the letter from which x_t is generated; also, let $F_t = 1$ be a tag indicating that a new phone segment starts at $t+1$. Given the constraint that $0 \leq n_i \leq 2$, for $0 \leq i \leq L_m$ and $0 \leq t \leq T_m$, the backwards messages $B_t(i)$ and $B_t^*(i)$ for the m^{th} training sample can be defined and computed as in Eq. 5.5 and Eq. 5.7. Note that for clarity we discard the index variable m in the derivation of the algorithm.

$$\begin{aligned} B_t(i) &\triangleq p(x_{t+1:T}|\mathbb{L}(x_t) = i, F_t = 1) \\ &= \sum_{j=i+1}^{\min\{L,i+1+U\}} B_t^*(j) \prod_{k=i+1}^{j-1} p(n_k = 0|\vec{l}_{i,\kappa}) \\ &= \sum_{j=i+1}^{\min\{L,i+1+U\}} B_t^*(j) \prod_{k=i+1}^{j-1} \phi_{\vec{l}_{i,\kappa}}(0) \end{aligned} \tag{5.5}$$

$$\begin{aligned} &B_t^*(i) \triangleq p(x_{t+1:T}|\mathbb{L}(x_{t+1}) = i, F_t = 1) \\ &= \sum_{d=1}^{T-t} p(x_{t+1:t+d}|\vec{l}_{i,\kappa}) B_{t+d}(i) \\ &= \sum_{d=1}^{T-t} \{ \sum_{c_{i,1}=1}^{K} \phi_{\vec{l}_{i,\kappa}}(1) \pi_{\vec{l}_{i,\kappa},1,1}(c_{i,1}) p(x_{t+1:t+d}|\theta_{c_{i,1}}) \\ &+ \sum_{v=1}^{d-1} \sum_{c_{i,1}}^{K} \sum_{c_{i,2}}^{K} \phi_{\vec{l}_{i,\kappa}}(2) \pi_{\vec{l}_{i,\kappa},2,1}(c_{i,1}) \pi_{\vec{l}_{i,\kappa},2,2}(c_{i,2}) \\ &\times p(x_{t+1:t+v}|\theta_{c_{i,1}}) p(x_{t+v+1:t+d}|\theta_{c_{i,2}}) \} B_{t+d}(i) \end{aligned} \tag{5.6, 5.7}$$

We use $x_{t_1:t_2}$ to denote the segment consisting of $x_{t_1}, \cdots, x_{t_2}$. Our inference algorithm only allows up to U letters to emit 0 acoustic units in a row. The value of U is set to 2 for our

Algorithm 5.4.1 Block-sample n_i and $c_{i,p}$ from $B_t(i)$ and $B_t^*(i)$

$i \leftarrow 0$
$t \leftarrow 0$
while $i < L \wedge t < T$ **do**
    $next_i \leftarrow SampleFromB_t(i)$
    if $next_i > i + 1$ **then**
        for $k = i + 1$ to $k = next_i - 1$ **do**
            $n_k \leftarrow 0$
        end for
    end if
    $d, n_i, \langle c_{i,p} \rangle, v \leftarrow SampleFromB_t^*(next_i)$
    $t \leftarrow t + d$
    $i \leftarrow next_i$
end while

experiments. $B_t(i)$ represents the total probability of all possible alignments between $x_{t+1:T}$ and $l_{i:L}$. $B_t^*(i)$ contains the probability of all the alignments between $x_{t+1:T}$ and $l_{i+1:L}$ that map x_{t+1} to l_i particularly. This alignment constraint between x_{t+1} and l_i is explicitly shown in the first term of Eq. 5.6, which represents how likely the speech segment $x_{t+1:t+d}$ is generated by l_i given l_i's context. This likelihood is simply the marginal probability of $p(x_{t+1:t+d}, n_i, c_{i,p}|\vec{l}_{i,\kappa})$ with n_i and $c_{i,p}$ integrated out, which can be expanded and computed as shown in the last three rows of Eq. 5.7. The index v specifies where the phone boundary is between the two acoustic units that l_i is aligned with when $n_i = 2$. Eq. 5.8 to Eq. 5.10 are the boundary conditions of the message passing algorithm. $B_0(0)$ carries the total probably of all possible alignments between $l_{1:L}$ and $x_{1:T}$. Eq. 5.9 specifies that at most U letters at the end of a sentence can be left unaligned with any speech features, while Eq. 5.10 indicates that all of the speech features in an utterance must be assigned to a letter.

$$B_0(0) = \sum_{j=1}^{\min\{L,U+1\}} B_0^*(j) \prod_{k=1}^{j-1} \phi_{\vec{l}_{i,\kappa}}(0) \tag{5.8}$$

$$B_T(i) \triangleq \begin{cases} 1 & \text{if } i = L \\ \prod_{j=i+1}^{L} \phi_{\vec{l_{i,\kappa}}}(0) & \text{if } L - U \leq i < L \\ 0 & \text{if } i < L - U \end{cases} \tag{5.9}$$

$$B_t(L) \triangleq \begin{cases} 1 & \text{if } t = T \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Given $B_t(i)$ and $B_t^*(i)$, n_i and $c_{i,p}$ for each letter in the utterance can be sampled using Alg. 5.4.1. The $SampleFromB_t(i)$ function in line 4 returns a random sample from the relative probability distribution composed by entries of the summation in Eq. 5.5. Line 5 to line 9 check whether l_i (and maybe l_{i+1}) is mapped to zero phonetic units. $next_i$ points to the letter that needs to be aligned with 1 or 2 phone segments starting from x_t. The number of phonetic units that l_{next_i} maps to and the identities of the units are sampled in $SampleFromB_t^*(i)$. This sub-routine generates a tuple of d, n_i, $\langle c_{i,p} \rangle$ as well as v (if $n_i = 2$) from all the entries of the summation shown in Eq. 5.7. Note that we use $\langle c_{i,p} \rangle$ to denote that $\langle c_{i,p} \rangle$ may consist of two numbers, $c_{i,1}$ and $c_{i,2}$, when $n_i = 2$.

■ 5.4.2 Heuristic Phone Boundary Elimination

The variables d and v, in Eq. 5.7, enumerate through every frame index in a sentence, treating each feature frame as a potential boundary between acoustic units. However, it is possible to exploit acoustic cues to avoid checking feature frames that are unlikely to be phonetic boundaries. We follow the pre-segmentation method described in Section 2.6 to skip roughly 80% of the feature frames and greatly speed up the computation of $B_t^*(i)$.

Another heuristic applied to our algorithm to reduce the search space for d and v is based on the observation that the average duration of phonetic units is usually no longer than 300 ms. Therefore, when computing $B_t^*(i)$, we only consider speech segments that are shorter than 300 ms to avoid aligning letters to speech segments that are too long to be phonetic units.

■ 5.4.3 Voice Activity Detection for Initializing a Silence Model

Before training the model, we apply an unsupervised voice activity detector based on [176, 175] to locate non-speech acoustic segments in the data. These non-speech segments are used to train a silence model, or a background noise model, which seeds one of the phonetic units to be discovered. In other words, we initialize one of the HMMs with the silence model instead

of generating its parameters from the prior. This pre-processing step aims to prevent the model from learning a large number of silence units as shown in Fig. 2.6.

■ 5.4.4 Sampling $\phi_{\vec{l}_\kappa}$, $\pi_{\vec{l}_\kappa,n_i,p}$, β and θ_c

Sampling $\phi_{\vec{l}_\kappa}$

To compute the posterior distribution of $\phi_{\vec{l}_\kappa}$, we count how many times $\vec{l}_\kappa$ is mapped to 0, 1 and 2 phonetic units from n_i^m. More specifically, we define $\mathcal{N}_{\vec{l}_\kappa}(j)$ for $0 \leq j \leq 2$ as follows:

$$\mathcal{N}_{\vec{l}_\kappa}(j) = \sum_{m=1}^{M} \sum_{i=1}^{L_m} \delta(n_i^m, j)\delta(\vec{l}_{i,\kappa}^m, \vec{l}_\kappa)$$

where we use $\delta(\cdot)$ to denote the discrete Kronecker delta. With $\mathcal{N}_{\vec{l}_\kappa}$, we can simply sample a new value for $\phi_{\vec{l}_\kappa}$ from the following distribution:

$$\phi_{\vec{l}_\kappa} \sim Dir(\eta + \mathcal{N}_{\vec{l}_\kappa})$$

Sampling $\pi_{\vec{l}_\kappa,n,p}$ and β

The posterior distributions of $\pi_{\vec{l}_\kappa,n,p}$ and β are constructed recursively due to the hierarchical structure imposed on $\pi_{\vec{l}_\kappa,n,p}$ and β. We start with gathering counts for updating the π variables at the lowest level, i.e., $\pi_{\vec{l}_2,n,p}$ given that κ is set to 2 in our model implementation, and then sample pseudo-counts for the π variables at higher hierarchies as well as β. With the pseudo-counts, a new β can be generated, which allows $\pi_{\vec{l}_\kappa,n,p}$ to be re-sampled sequentially.

More specifically, we define $\mathcal{C}_{\vec{l}_2,n,p}(k)$ to be the number of times that $\vec{l}_2$ is mapped to n units and the unit in position p is the k^{th} phonetic unit. This value can be counted from the current values of $c_{i,p}^m$ as follows.

$$\mathcal{C}_{\vec{l}_2,n,p}(k) = \sum_{m=1}^{M} \sum_{i=1}^{L_m} \delta(\vec{l}_{i,2}, \vec{l}_2)\delta(n_i^m, n)\delta(c_{i,p}^m, k)$$

To derive the posterior distribution of $\pi_{\vec{l}_1,n,p}$ analytically, we need to sample pseudo-counts $\mathcal{C}_{\vec{l}_1,n,p}$, which is defined as follows.

$$\mathcal{C}_{\vec{l}_1,n,p}(k) = \sum_{\vec{l}_2 \in \mathcal{U}_{\vec{l}_1}} \sum_{i=1}^{\mathcal{C}_{\vec{l}_2,n,p}(k)} \mathbb{I}[\nu_i < \frac{\alpha_2 \pi_{\vec{l}_1,n,p}(k)}{i + \alpha_2 \pi_{\vec{l}_1,n,p}(k)}] \tag{5.11}$$

We use $\mathcal{U}_{\vec{l}_1} = \{\vec{l}_2 | \mathbb{P}(\vec{l}_2) = \vec{l}_1\}$ to denote the set of $\vec{l}_2$ whose parent is $\vec{l}_1$ and ν_i to represent random variables sampled from a uniform distribution between 0 and 1. Eq. 5.11 can be applied recursively to compute $\mathcal{C}_{\vec{l}_0,n,p}(k)$ and $\mathcal{C}_{\_,n,p}(k)$, the pseudo-counts that are applied to the conjugate priors of $\pi_{\vec{l}_0,n,p}$ and β. With the pseudo-count variables computed, new values for β and $\pi_{\vec{l}_\kappa,n,p}$ can be sampled sequentially as shown in Eq. 5.12 to Eq. 5.14.

$$\beta \sim Dir(\gamma + \mathcal{C}_{\_,n,p}) \tag{5.12}$$

$$\pi_{\vec{l}_\kappa,n,p} \sim Dir(\alpha_\kappa \beta + \mathcal{C}_{\vec{l}_\kappa,n,p}) \text{ for } \kappa = 0 \tag{5.13}$$

$$\pi_{\vec{l}_\kappa,n,p} \sim Dir(\alpha_\kappa \pi_{\vec{l}_{\kappa-1},n,p} + \mathcal{C}_{\vec{l}_\kappa,n,p}) \text{ for } \kappa \geq 1 \tag{5.14}$$

Sampling θ_c

Finally, after assigning the sub-word unit label $c_{i,p}$ to each speech segment, we block-sample the state and mixture id for each feature frame within the speech segment using the HMM associated with the sub-word label $\theta_{c_{i,p}}$. From the state and mixture assignment of each feature vector, we can collect relevant counts to update the priors for the transition matrix and the state emission distributions of each HMM. New parameters of θ_c can then be generated from the updated priors. See Section 2.5 for a more detailed description for sampling new values for the HMM parameters.

■ 5.5 Automatic Speech Recognition Experiments

To test the effectiveness of our model for joint learning phonetic units and word pronunciations from an annotated speech corpus, we construct speech recognizers out of the training results of our model. The performance of the recognizers is evaluated and compared against three baselines: first, a grapheme-based speech recognizer; second, a recognizer built by using an expert-crafted lexicon, which is referred to as an expert lexicon in the rest of the paper for simplicity; and third, a recognizer built by discovering the phonetic units and L2S pronunciation rules *sequentially* without using a lexicon. In this section, we provide a detailed description of the experimental setup.

■ 5.5.1 Jupiter Corpus

All the speech recognition experiments reported in this paper are performed on a weather query dataset, which consists of narrow-band, conversational telephone speech [195]. We evaluate

η	γ	α_0	α_1	α_2	θ_0	κ	K
$\langle 0.1 \rangle_3$	$\langle 10 \rangle_{100}$	1	0.1	0.2	*	2	100

Table 5.1. The values of the hyperparameters of our model. We use $\langle a \rangle_D$ to denote a D-dimensional vector with all entries being a. *We follow the procedure reported in Section 2.4 to set up the HMM prior θ_0.

our model on this weather query corpus because previous work [125] also employed the same dataset to investigate a stochastic lexicon learning scheme, which can be integrated into our learning framework as shown later in this section. We follow the experimental setup of [125] and split the corpus into a training set of 87,351 utterances, a dev set of 1,179 utterances and a test set of 3,497 utterances. A subset of 10,000 utterances is randomly selected from the training set. We use this subset of data for training our model to demonstrate that our model is able to discover the phonetic composition and the pronunciation rules of a language even from just a few hours of data.

■ 5.5.2 Building a Recognizer from Our Model

The values of the hyperparameters of our model are listed in Table 5.1. We run the inference procedure described in Section 5.4 for 10,000 times on the randomly selected 10,000 utterances. The samples of $\phi_{\vec{l}_\kappa}$ and $\pi_{\vec{l}_\kappa n,p}$ from the last iteration are used to decode n_i^m and $c_{i,p}^m$ for each sentence in the entire training set by following the block-sampling algorithm described in Section 5.4.1. Since $c_{i,p}^m$ is the phonetic mapping of l_i^m, by concatenating the phonetic mapping of every letter in a word, we can obtain a pronunciation of the word represented in the labels of discovered phonetic units. For example, assume that word w appears in sentence m and consists of $l_3 l_4 l_5$ (the sentence index m is ignored for simplicity). Also, assume that after decoding, $n_3 = 1$, $n_4 = 2$ and $n_5 = 1$. A pronunciation of w is then encoded by the sequence of phonetic labels $c_{3,1} c_{4,1} c_{4,2} c_{5,1}$. By repeating this process for each word in every sentence for the training set, a list of word pronunciations can be compiled and used as a stochastic lexicon to build a speech recognizer.

In theory, the HMMs inferred by our model can be directly used as the acoustic model of a monophone speech recognizer. However, if we regard the $c_{i,p}$ labels of each utterance as the phone transcription of the sentence, then a new acoustic model can be easily re-trained on the entire training data set. More conveniently, the phone boundaries corresponding to the $c_{i,p}$ labels are the by-products of the block-sampling algorithm, which are indicated by the values of d and v in line 10 of Alg. 5.4.1 and can be easily saved during the sampling procedure. Since these data are readily available, we re-build a context-independent model on the entire data

set. In this new acoustic model, a 3-state HMM is used to model each phonetic unit, and the emission probability of each state is modeled by a 32-mixture GMM.

Finally, a trigram language model is built by using the word transcriptions in the full training set. This language model is utilized in all speech recognition experiments reported in this paper. The MIT SUMMIT speech recognition system [193] and the MIT Finite State Transducer (FST) toolkit [68] are used to build all the recognizers used in this study. With the language model, the lexicon and the context-independent acoustic model constructed by the methods described in this section, we can build a speech recognizer from the learning output of the proposed model without the need of a pre-defined phone inventory and any expert-crafted lexicons.

Pronunciation Mixture Model Retraining

The authors of [125] presented the Pronunciation Mixture Model (PMM) for composing stochastic lexicons that outperform pronunciation dictionaries created by experts. Although the PMM framework was designed to incorporate and augment expert lexicons, we found that it can be adapted to polish the pronunciation list generated by our model. In particular, the training procedure for PMMs includes three steps. First, train a L2S model from a manually specified expert-pronunciation lexicon; second, generate a list of pronunciations for each word in the dataset using the L2S model; and finally, use an acoustic model to re-weight the pronunciations based on the acoustic scores of the spoken examples of each word.

To adapt this procedure for our purposes, we simply plug in the word pronunciations and the acoustic model generated by our model. Once we obtain the re-weighted lexicon, we re-generate forced phone alignments and retrain the acoustic model, which can be utilized to repeat the PMM lexicon re-weighting procedure. For our experiments, we iterate through this model refining process until the recognition performance converges.

Triphone Model

Conventionally, to train a context-dependent acoustic model, a list of questions based on the linguistic properties of phonetic units is required for growing decision tree classifiers [188]. However, such language-specific knowledge is not available for our training framework; therefore, our strategy is to compile a question list that treats each phonetic unit as a unique linguistic class. In other words, our approach to training a context-dependent acoustic model for the automatically discovered units is to let the decision trees grow fully based on acoustic evidence.

5.5.3 Baseline Systems

We compare the recognizers trained by following the procedures described in Section 5.5.2 against three baselines. The first baseline is a grapheme-based speech recognizer. We follow the procedure described in [91] and train a 3-state HMM for each grapheme, which we refer to as the *monophone grapheme* model. Furthermore, we create a *singleton question set* [91], in which each grapheme is listed as a question, to train a *triphone grapheme* model. Note that to enforce better initial alignments between the graphemes and the speech data, we use a pre-trained acoustic model to identify the non-speech segments at the beginning and the end of each utterance before starting training the monophone grapheme model.

Our model jointly discovers the phonetic inventory and the L2S mapping rules from a set of transcribed data. An alternative of our approach is to learn the two latent structures sequentially. We use the DPHMM model described in Chapter 2 to learn a set of acoustic models from the speech data and use these acoustic models to generate a phone transcription for each utterance. The phone transcriptions along with the corresponding word transcriptions are fed as inputs to the L2S model proposed in [11]. A stochastic lexicon can be learned by applying the L2S model and the discovered acoustic models to PMM. This two-stage approach for training a speech recognizer without an expert lexicon is referred to as the *sequential model* in this paper.

Finally, we compare our system against a recognizer trained from an *oracle* recognition system. We build the oracle recognizer on the same weather query corpus by following the procedure presented in [125]. This oracle recognizer is then applied to generate forced-aligned phone transcriptions for the training utterances, from which we can build both monophone and triphone acoustic models. Note that for training the triphone model, we compose a singleton question list [91] that has every expert-defined phonetic unit as a question. We use this singleton question list instead of a more sophisticated one to ensure that this baseline and our system differ only in the acoustic model and the lexicon used to generate the initial phone transcriptions. We call this baseline the *oracle* baseline.

5.6 Results and Analysis

5.6.1 Analysis on the Discovered Letter-to-sound Mapping Rules

Before showing the recognition results, we analyze the learning behavior of our model in this section. We are particularly interested in knowing whether the model infers any useful letter-to-sound mapping rules from the dataset that correspond to our understanding of the relationship between the written and spoken English. To gain some insight, we define and compute $p(z|\vec{l}_\kappa)$

from the learning output of our model as follows.

$$p(z|\vec{l_\kappa}) = \sum_{c=1}^{K} p(z, c|\vec{l_\kappa}) = \sum_{c=1}^{K} p(c|\vec{l_\kappa})p(z|c) \tag{5.15}$$

where $z \in Z$ is a standard English phone, and Z represents the expert-defined phone inventory used in this analysis. The set of 47 phones contained in Z is shown at the bottom of Fig. 5.3. The probability $p(z|\vec{l_\kappa})$ indicates how likely it is to map the center letter of $\vec{l_\kappa}$ within the specific context implied by $\vec{l_\kappa}$ to the standard phone z. The value of $p(z|\vec{l_\kappa})$ can be computed by marginalizing over $c = 1, \cdots, K$ for $p(z, c|\vec{l_\kappa})$, which can be further decomposed into $p(c|\vec{l_\kappa})p(z|c)$ as shown in Eq. 5.15. The first term $p(c|\vec{l_\kappa})$ is the probability of mapping the letter sequence $\vec{l_\kappa}$ to the automatically induced phonetic unit c, which can be calculated by using the inferred model variables $\phi_{\vec{l_\kappa}}$ and $\pi_{\vec{l_\kappa},n,p}$ as follows.

$$p(c|\vec{l_\kappa}) = \{\underbrace{\phi_{\vec{l_\kappa}}(1)\pi_{\vec{l_\kappa},1,1}(c)}_{\text{(a)}} + \underbrace{\phi_{\vec{l_\kappa}}(2)(\pi_{\vec{l_\kappa},2,1}(c) + \pi_{\vec{l_\kappa},2,2}(c) - \pi_{\vec{l_\kappa},2,1}(c)\pi_{\vec{l_\kappa},2,2}(c))}_{\text{(b)}}\} \tag{5.16}$$

where Eq. 5.16-(a) and Eq. 5.16-(b) stand for the probabilities of mapping the center letter of $\vec{l_\kappa}$ to c when $\vec{l_\kappa}$ is aligned with one and two units respectively.

The second term $p(z|c)$ of Eq. 5.15 represents the likelihood of mapping the induced phonetic unit c to the standard English phone z. We compute $p(z|c)$ by aligning the phonetic transcriptions generated by our model ($c_{i,p}$) to those produced by a speech recognizer (z_j) and calculate the overall overlap ratio between c and each standard phone unit. An example of calculating $p(z|c)$ for the discovered unit 54 is illustrated in Fig. 5.2, in which we assume that unit 54 only appears twice in the corpus. To compute $p(z|54)$, we first examine the time-stamped alignments between the model-inferred and recognizer-produced phone transcriptions and extract all instances that involve unit 54 as depicted in (i) and (ii) of Fig. 5.2-(a). The variable t in Fig. 5.2-(a) specifies the time indices of the boundaries of unit 54 and the aligned English phones. We compute the overlap ratio between unit 54 and each of the underlying English standard phones, and then normalize these overlap ratios to obtain $p(z|54)$ as shown in Figs. 5.2-(a) and (b).

To visualize the letter-to-sound mapping rules discovered by our model, we compute $p(z|\vec{l_0})$ using Eq. 5.16, which stands for the probability of mapping the letter l to the English phone z

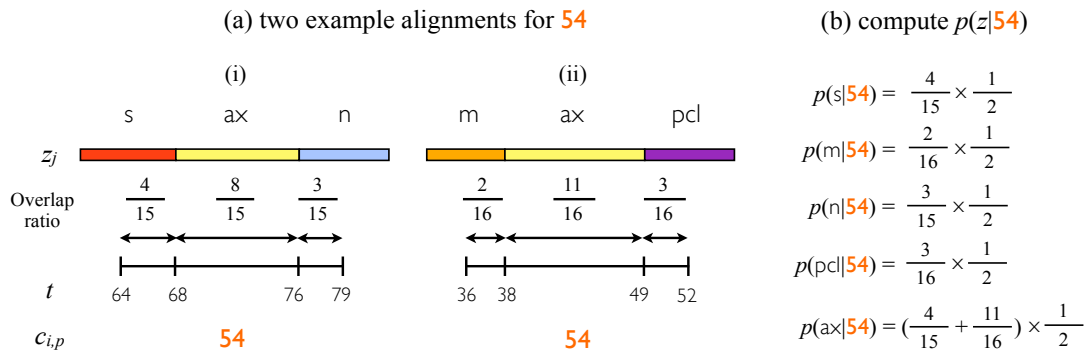


Figure 5.2. An illustration of the computation of $p(z|c)$, the mapping probabilities between the automatically induced units and the standard phones. The discovered unit 54 is used in this illustration. (a) shows two alignments between the unit 54 and the standard phones. The variable t indicates the time indices of the boundaries of each phonetic unit. (b) explains how to compute $p(z|54)$ by normalizing the overlap ratios between unit 54 and all the aligned standard phone units.

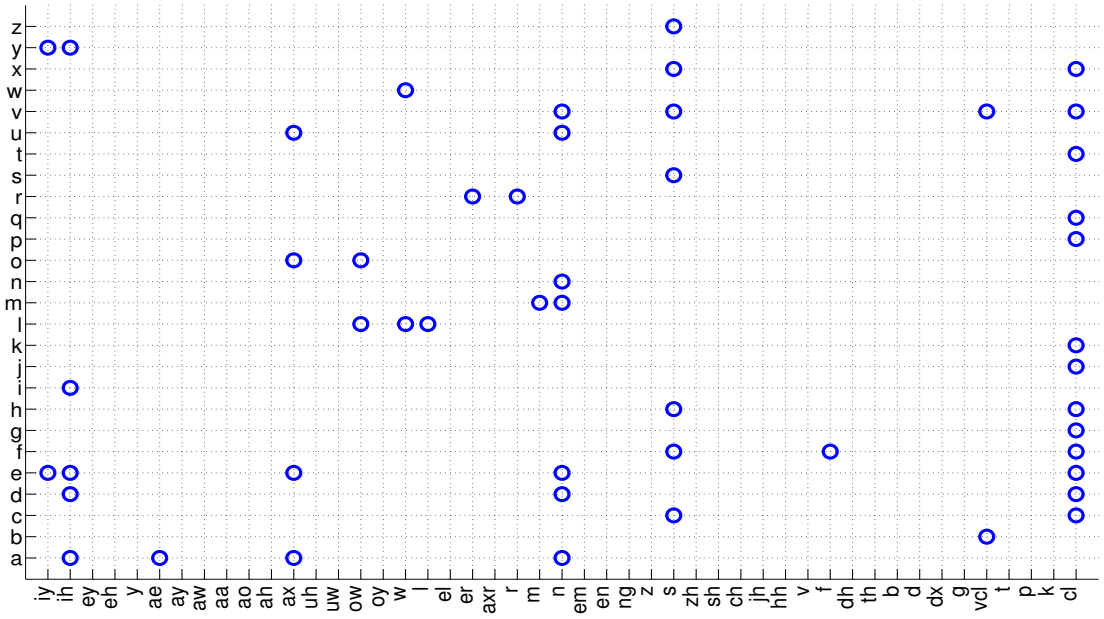


Figure 5.3. Visualization of the letter-to-sound mapping for English produced my our model.

without considering any specific context. Based on the results, we pair each letter l with the phone z that achieves the highest value of $p(z|\vec{l}_0)$ (and possibly with other phones that have

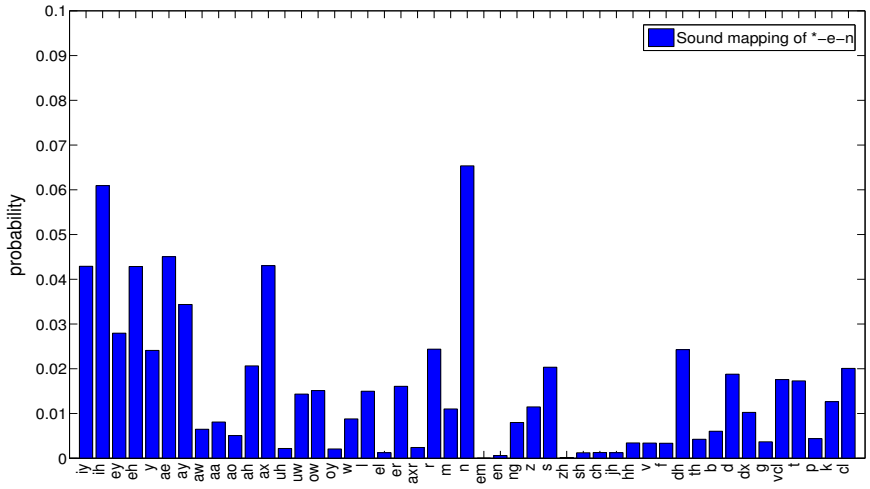


Figure 5.4. The probabilistic letter-to-sound mapping distribution our model learns for the letter ***e*** followed by the letter ***n***.

equally high probabilities) and compose the mapping matrix between the 26 English letters and the 47 English phone. We present the matrix in Fig. 5.3 and discuss some observations.

First, the matrix seems to suggest that no letters are mapped to the releases of the stop consonants, /b/, /d/, /g/, /p/, /t/, and /k/. However, this is not a surprising result because the duration of a stop release tends to be shorter than that of a stop closure, which causes $p(z|c)$ to be much smaller for the release phone units than for the closure units /vcl/ and /cl/. Given that we only align each letter to the phone that scores the highest $p(z|\vec{l_0})$, the letters that encode the stop sounds, *b*, *d*, *g*, *p*, *t*, *k*, and *q*, are mostly only mapped to the closure units in the mapping matrix.

As shown in Fig. 5.3, the letters *a*, *e*, and *u* are mapped to not only vowels but also the nasal sound /n/, which is quite unexpected. Nevertheless, with a more careful examination on the alignments, we find that when the letters *a*, *e*, and *u* are followed by the letter *n*, our model tends to align the letters *a*, *e*, *u* to both a vowel, which is usually reduced to a schwa, and the following consonant /n/. This observation can be illustrated by the average phone mapping distribution $p(z|\vec{l_1})$ for $z \in Z$ over the set of letter sequences $\vec{l_1} \in \{* - e - n\}$, in which the letter *e* is followed by the letter *n*. The distribution is presented in Fig. 5.4, from which we can see that the letter *e* is usually mapped to the short vowel /ih/ and the consonant /n/.

In spite of these less expected mapping behaviors, many plausible letter-to-sound mappings are shown in Fig. 5.3. For example, the letter *w* is accurately mapped to the semi-vowel /w/, and

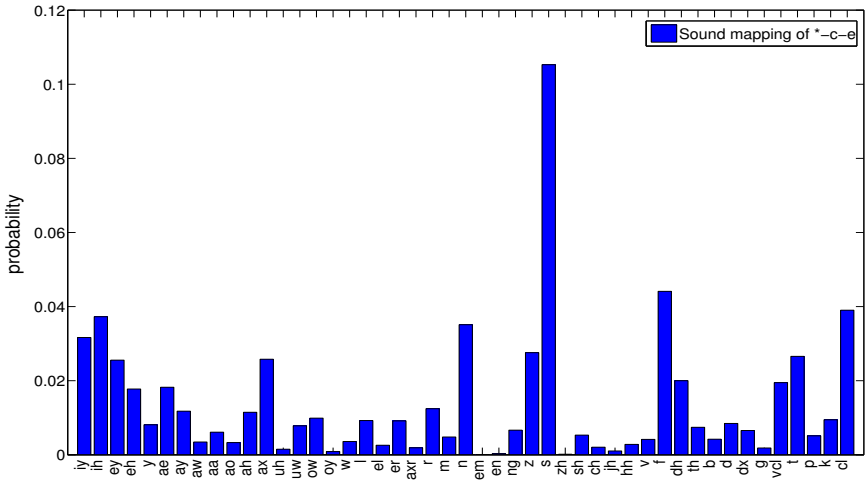


Figure 5.5. The probabilistic letter-to-sound mapping distribution our model learns for the letter ***c*** followed by the letter ***e***.

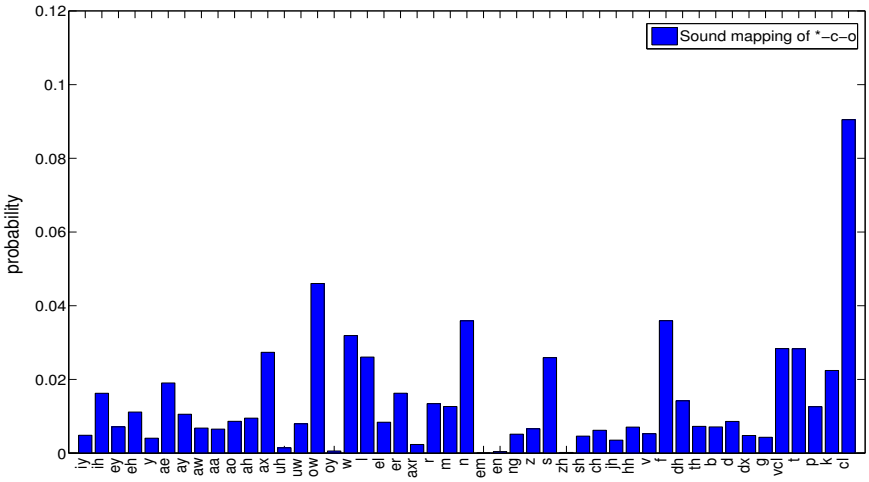


Figure 5.6. The probabilistic letter-to-sound mapping distribution our model learns for the letter ***c*** followed by the letter ***o***.

the letter *l* is assigned to the back vowel /ow/, the semi-vowels /w/ and /l/, which are acoustically similar. In addition, a closer look at the phone correspondence for the letter *x* demonstrates that the model successfully captures that the letter *x* encodes two sounds: one fricative consonant /s/ and one stop consonant /cl (k)/. Furthermore, the mapping matrix also reveals that our model precisely learns that the letter *c* can be pronounced as /s/ and /cl (k)/.

Word	Discovered pronunciation
Russia	28 17 34 47 70
Scotia	24 72 79 34 47 77
Shanghai	34 16 22 61 12 38
Champaign	34 16 91 35 26 2 22

Table 5.2. Examples of words that contain the sound /sh/. The /sh/ sound is encoded in many ways in English such as the **ss** in *Russia*, the **ti** in *Scotia*, the **sh** in *Shanghai*, and the **Ch** in *Champaign*. Nevertheless, our model can consistently map these encoding variations for /sh/ to an automatically discovered phonetic unit 34 that represents the consonant /sh/.

To investigate the last observation further, we average $p(z|\vec{l}_1)$ over two sets of letter sequences, $\vec{l}_1 \in \{* - c - e\}$ and $\vec{l}_1 \in \{* - c - o\}$, respectively to study the sound mapping probabilistic distributions for the letter c within two different contexts. In the first context, the letter c precedes the letter e, while in the second context, the letter c is followed by the letter o. The two probabilistic distributions are shown in Fig. 5.5 and Fig. 5.6 respectively. From Fig. 5.5, we can see that the model accurately infers the mapping between the letter c and the sound /s/ when the letter c precedes the letter e. In addition, Fig. 5.6 shows that the model correctly learns the pronunciation of c should be a stop sound when followed by the letter o.

Overall speaking, the letter-to-sound mapping rules our model infers do not always correspond to those defined by experts. However, as shown by many mapping pairs in Fig. 5.3, our model indeed discovers informative relationships between the written and spoken forms of English in a fully unsupervised manner. More importantly, the comparison between Fig. 5.5 and Fig. 5.6 clearly shows the capability of our model in inducing context-dependent letter pronunciations, which demonstrates the strength of the hierarchical design of our model.

■ 5.6.2 Interpretation of the Discovered Pronunciations

As described in Section 5.5, the word pronunciations discovered by our model are represented as sequences of automatically learned phonetic units. Unlike expert-defined phones, these induced units are denoted by integers and tend to be difficult to interpret. Therefore, in this section, we examine different ways to shed light on the word pronunciations our model learns and try to gain intuition behind the inferred sequences of integers.

First, we present the word pronunciations that our model finds for *Russia*, *Scotia*, *Shanghai*, and *Champaign*. One thing that these four words share in common is that they all contain the /sh/ sound. However, all of the four words encode this sound in a different way. The letters

Source	Target
10 6 98 85 69 22 3 83	f l ih n tcl t
10 66 28 92 25 31 49 42	f er n ae n dcl d ow
27 47 7 22 9 19 28 37 3 34	kcl k ey m bcl b r ih dcl jh

Table 5.3. Training examples in the parallel corpus for building the translation system. The parallel corpus is composed of automatically discovered word pronunciations and expert-defined pronunciations, which are treated as sentences from the source and the target language respectively.

Word	automatically-induced pronunciation	Translated automatically-induced pronunciation	Expert-defined pronunciation
Jakarta	34 47 35 19 66 56 17 55 86 11 70	dcl jh ae kcl k aa r tcl t ax	dcl jh ax kcl k aa r tcl t ax
Barbara	42 56 67 42 66 56 70	bcl b aa r bcl b r ax	bcl b aa r bcl b r ax
Marseille	9 39 11 24 69 29 73 77	m aa r s ey ax	m aa r s ey l
mountain	39 79 45 55 3 94 22 33	m ow n tcl t ax n	m aw n tcl t ax n
Borneo	42 97 43 46 20 16 79 88	bcl b ao r n y ax l	bcl b ao r n iy ow
flights	10 40 12 59 3 93	f l ay tcl t s	f l ay tcl t s

Table 5.4. Automatically inferred word pronunciations and their translations denoted in expert-defined phonetic units, which are generated by using a Moses translation system. The corresponding expert-defined pronunciation for each word is listed in the right-most column.

corresponding to the /sh/ sound in each of the four words are highlighted in Table 5.2. In spite of the variations observed in the spellings, our model consistently maps the different encodings for /sh/ to the phonetic unit 34 as shown in Table 5.2. In addition, by comparing the pronunciations associated with the letter *s* in both *Scotia* and *Shanghai*, we can see that our model is able to infer different pronunciations for the two *s*'s that are in different contexts. The examples in Table 5.2 show that our model can effectively handle irregularities that exist in the mapping between the written system and the spoken system of a language.

Furthermore, in order to interpret the learned word pronunciations, we train a translation system using Moses, an open source toolkit for statistical machine translation [94]. More specifically, we exploit 10,000 word pronunciations discovered by our model (including multiple entries for a word) and the corresponding pronunciations defined by experts to create a parallel corpus. We treat the automatically induced pronunciations as sentences in the source language and the expert-defined pronunciations as sentences in the target language for training

unit(%)	Monophone
Our model	17.0
Oracle	13.8
Grapheme	32.7
Sequential model	31.4

Table 5.5. Word error rates generated by the four monophone recognizers described in Sec. 5.5.2 and Sec. 5.5.3 on the weather query corpus.

the translation system. Some training examples in the parallel corpus are shown in Table 5.3. We build a translation system with this parallel corpus by using the default setup described in the project webpage of Moses [2].

With this translation system, we can convert the automatically discovered word pronunciations to pronunciations that are labelled by expert-defined phone units, which are more comprehensible, and allows us to gain insights into the induced phone sequences. Some of the translation results along with the corresponding expert-defined pronunciations are presented in Table 5.4 (more translation examples can be found in Appendix A). While the translated pronunciations do not always match the expert-defined ones perfectly, we can still see that the inferred pronunciations indeed carry useful information annotating the underlying sound sequences of the words. With the intuitions drawn from Table 5.2 and Table 5.4, we now present the performance of the speech recognizers built with the induced pronunciation dictionary.

■ 5.6.3 Monophone Systems

Table 5.5 shows the Word Error Rates (WERs) produced by the four monophone recognizers described in Sec. 5.5.2 and Sec. 5.5.3. It can be seen that our model outperforms the grapheme and the sequential model baselines significantly while approaching the performance of the supervised oracle baseline. The improvement over the sequential baseline demonstrates the strength of the proposed joint learning framework. More specifically, unlike the sequential baseline, in which the acoustic units are discovered independently from the text data, our model is able to exploit the L2S mapping constraints provided by the word transcriptions to cluster speech segments.

By comparing our model to the grapheme baseline, we can see the advantage of modeling the pronunciations of a letter using a mixture model, especially for a language like English

[2]Moses baseline: http://www.statmt.org/moses/?n=Moses.Baseline

which has many pronunciation irregularities. However, even for languages with straightforward pronunciation rules, the concept of modeling letter pronunciations using mixture models still applies. The main difference is that the mixture weights for letters of languages with simple pronunciation rules will be sparser and spikier. In other words, in theory, our model should always perform comparable to, if not better than, grapheme recognizers.

Last but not least, the recognizer trained with the automatically induced lexicon performs comparably to the recognizer initialized by an oracle recognition system, which demonstrates the effectiveness of the proposed model for discovering the phonetic inventory and a pronunciation lexicon from an annotated corpus. In the next section, we provide some insights into the quality of the learned lexicon and into what could have caused the performance gap between our model and the conventionally trained recognizer.

■ 5.6.4 Pronunciation Entropy

The major difference between the recognizer that is trained by using our model and the recognizer that is seeded by an oracle recognition system is that the former uses an automatically discovered lexicon, while the latter exploits an expert-defined pronunciation dictionary. In order to quantify, as well as to gain insights into, the difference between these two lexicons, we define the average pronunciation entropy $\hat{H}$ of a lexicon as follows.

$$\hat{H} \equiv \frac{-1}{|V|} \sum_{w \in V} \sum_{b \in \mathcal{B}(w)} p(b) \log p(b) \tag{5.17}$$

where V denotes the vocabulary of a lexicon, $\mathcal{B}(w)$ represents the set of pronunciations of a word w and $p(b)$ stands for the weight of a certain pronunciation b. Intuitively, we can regard $\hat{H}$ as an indicator of how much pronunciation variation that each word in a lexicon has on average. Table 5.6 shows that the $\hat{H}$ values of the lexicon induced by our model and the expert-defined lexicon as well as their respective PMM-refined versions. Note that we build the PMM-refined version of the expert-defined lexicon by following the L2P-PMM framework described in [125]. In Table 5.6, we can see that the automatically-discovered lexicon and its PMM-reweighted versions have much higher $\hat{H}$ values than their expert-defined counterparts. These higher $\hat{H}$ values imply that the lexicon induced by our model contains more pronunciation variation than the expert-defined lexicon. Therefore, the lattices constructed during the decoding process for our recognizer tend to be larger than those constructed for the oracle baseline. The larger lattices imply less constrained search space, which could help explain the performance gap between the two systems in Table 5.5 and Table 5.6.

Our model	*PMM iterations*		
(Discovered lexicon)	0	1	2
$\hat{H}$ (bit)	4.58	3.47	3.03
WER (%)	17.0	16.6	15.9
Oracle	*PMM iterations*		
(Expert lexicon)	0	1	2
$\hat{H}$ (bit)	0.69	0.90	0.92
WER (%)	13.8	12.8	12.4

Table 5.6. The upper-half of the table shows the average pronunciation entropies, $\hat{H}$, of the lexicons induced by our model and refined by PMM as well as the WERs of the monophone recognizers built with the corresponding lexicons for the weather query corpus. The definition of $\hat{H}$ can be found in Sec. 5.6.4. The first row of the lower-half of the table lists the average pronunciation entropies, $\hat{H}$, of the expert-defined lexicon and the lexicons generated and weighted by the L2P-PMM framework described in [125]. The second row of the lower-half of the table shows the WERs of the recognizers that are trained with the expert-lexicon and its PMM-refined versions.

pronunciations	*pronunciation probabilities*		
	Our model	1 PMM	2 PMM
93 56 87 39 19	0.125	-	-
93 56 61 87 73 99	0.125	-	-
11 56 61 87 73 99	0.125	0.400	0.419
93 20 75 87 17 27 52	0.125	0.125	0.124
55 93 56 61 87 73 84 19	0.125	0.220	0.210
93 26 61 87 49	0.125	0.128	0.140
63 83 86 87 73 53 19	0.125	-	-
93 26 61 87 61	0.125	0.127	0.107

Table 5.7. Pronunciation lists of the word *Burma* produced by our model and refined by PMM after 1 and 2 iterations.

As shown in Table 5.6, even though the lexicon induced by our model is noisier than the expert-defined dictionary, the PMM retraining framework consistently refines the induced lexicon and improves the performance of the recognizers. To the best of our knowledge, we are the first to apply PMM to lexicons that are created by a fully unsupervised method. Therefore, in this chapter, we provide further analysis on how PMM helps enhance the performance of our model.

Unit(%)	Triphone
Our model	13.4
Oracle	10.0
Grapheme	15.7

Table 5.8. Word error rates of the triphone recognizers. The triphone recognizers are all built by using the phone transcriptions generated by their best monohpone system. For the oracle initialized baseline and for our model, the PMM-refined lexicons are used to build the triphone recognizers.

■ 5.6.5 Pronunciation Refinement by PMM

We compare the pronunciation lists for the word *Burma* generated by our model and refined iteratively by PMM in Table 5.7. The first column of Table 5.7 shows all the pronunciations of *Burma* discovered by our model. While it is possible to assign probabilities proportional to the decoding scores, we assign equal probabilities to all the learned pronunciations to create the stochastic list. As demonstrated in the third and the fourth columns of Table 5.7, the PMM framework is able to iteratively re-distribute the pronunciation weights and filter out less-likely pronunciations, which effectively reduces both the size and the entropy of the stochastic lexicon generated by our model. The benefits of using the PMM to refine the induced lexicon are twofold. First, the search space constructed during the recognition decoding process with the refined lexicon is more constrained, which is the main reason why the PMM is capable of improving the performance of the monophone recognizer that is trained with the output of our model. Secondly, and more importantly, the refined lexicon can greatly reduce the size of the FST built for the triphone recognizer of our model. These two observations illustrate why the PMM framework can be an useful tool for enhancing the lexicon discovered automatically by our model.

■ 5.6.6 Triphone Systems

The best monophone systems of the grapheme baseline, the oracle baseline and our model are used to generate forced-aligned phone transcriptions, which are used to train the triphone models described in Sec. 5.5.2 and Sec. 5.5.3. Table 5.8 shows the WERs of the triphone recognition systems. Note that if a more conventional question list, for example, a list that contains rules to classify phones into different broad classes, is used to build the oracle triphone system [188], the WER can be reduced to 6.5%. However, as mentioned earlier, in order to gain insights into the quality of the induced lexicon and the discovered phonetic set, we compare our

model against an oracle triphone system that is built by using a singleton question set.

By comparing Table 5.5 and Table 5.8, we can see that the grapheme triphone improves by a large margin compared to its monophone counterpart, which is consistent with the results reported in [91]. However, even though the grapheme baseline achieves a great performance gain with context-dependent acoustic models, the recognizer trained using the lexicon learned by our model and subsequently refined by PMM still outperforms the grapheme baseline. The consistently better performance our model achieves over the grapheme baseline demonstrates the strength of modeling the pronunciation of each letter with a mixture model that is presented in this chapter.

Last but not least, by comparing Table 5.5 and Table 5.8, it can be seen that the relative performance gain achieved by our model is similar to that obtained by the oracle baseline. Both Table 5.5 and Table 5.8 show that even without exploiting any language-specific knowledge during training, our recognizer is able to perform comparably with the recognizer trained using an expert lexicon. The ability of our model to obtain such similar performance further supports the effectiveness of the joint learning framework proposed in this chapter for discovering the phonetic inventory and the word pronunciations from simply an annotated speech corpus.

■ 5.7 Chapter Conclusion

In this chapter, we present a hierarchical Bayesian model for simultaneously discovering acoustic units and learning word pronunciations from transcribed spoken utterances. Both monophone and triphone recognizers can be built on the discovered acoustic units and the inferred lexicon. When tested on an English corpus of spontaneous weather queries, the recognizers trained with the proposed unsupervised method consistently outperform grapheme-based recognizers and approach the performance of recognizers trained with expert-defined lexicons.

Our work can be regarded as the first step towards untangling the dependence on expert-defined lexicons for training ASR systems. We believe that automatic induction of the encoding paradigm between the written and spoken systems of a language is the key to greatly increasing the multilingual capacity of speech recognizers. One interesting related research problem, which is beyond the scope of this thesis, is the even more daunting task of learning units and pronunciations from non-parallel text and speech data, which has similarities to a decipherment task [58]. Since collecting non-parallel speech and text data of a language is much easier than creating a transcribed speech corpus, the ability to acquire a pronunciation dictionary from non-parallel data represents a tremendous potential for extending the speech

recognition capacity to much more languages spoken in the world. With the recent success of deciphering mysterious codes, lost languages, and learning a translation lexicon from non-parallel data [93, 92, 157, 171, 26] using computational decryption methods, we believe that unsupervised discovery of pronunciation lexicons from non-parallel speech and text data is a promising research area.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we pose the challenge of discovering linguistic structures from speech signals, and investigate nonparametric Bayesian methods to tackle this challenge. The experimental results demonstrate that by imposing a Dirichlet process prior on HMMs that are used to model phonetic units, our DPHMM model can discover sub-word units that are highly correlated with standard phones defined by experts. Some of the discovered units even carry useful contextual information. Furthermore, by leveraging the intrinsic clustering property of Pitman-Yor processes, which encourages parameter sharing, our integration of adaptor grammars, noisy-channel model, and acoustic model is shown to be able to capture recurrent acoustic patterns that correspond to frequent syllabic and lexical units. We are the first to apply adaptor grammars to non-symbolic data, and the experimental results reveal that a noisy-channel model is necessary for absorbing the confusability that often exists in non-symbolic input.

In addition to exploring computational models for inferring latent structures in speech signals, we also apply the induced structures to an essential task of the language acquisition process: one-shot learning of spoken words. The experimental results indicate that learning the fundamental representation of a language, i.e., the phonetic inventory, plays an important role in recognizing and synthesizing new spoken words from just one or a few examples. Finally, we go beyond just speech data and invent a framework that automatically discovers the connection between the written-form and the spoken-form of a language. We utilize this framework to learn a fully data-driven pronunciation dictionary, which enables the development of well-performing speech recognizers without any expert knowledge.

6.2 Future Work

Several avenues of investigation arise from the work presented in this thesis. In addition to those discussed at the end of each chapter, we describe some more future research directions below.

6.2.1 A Data-driven approach to Formalizing Phonological Knowledge

Conventionally, researchers apply rule-based knowledge to define and study the phonological properties of a language. In contrast, the DPHMM model proposed in Chapter 2 provides a fully data-driven approach to formalizing these linguistic properties. This automatic approach implies that we can bypass the need of arbitrarily deciding the phonemes, phones, and allophonic variations of a language, and define these phonological structures by exploiting evidence from speech data of the language. In particular, the DPHMM model can be used as a tool to help linguists to study the phonology of a language. For example, we can first employ the DPHMM model to learn an initial phone set that captures detailed context-dependent acoustic variations in the individual phonetic units, and then have linguists refine or merge the distinct phonetic units by listening to the associated speech segments. We envision that a hybrid technique that combines knowledge-based and data-driven approaches would be the key to learning the phonological structures for other languages in the world, especially for low-resource languages.

6.2.2 Integrating Prior Knowledge for Learning

The training of most of the models presented in this thesis is fairly unsupervised. More specifically, we do not impose any language-specific knowledge on the models; instead, we only rely on the weak constraints implied by the model priors during learning. The rationale behind this design is that language-specific knowledge is not always accessible; therefore, the learning of the models should not depend on it. While language-specific knowledge may be difficult to obtain, some universal rules are readily observed and available. Take universal phonetics as an example. The vowels /i/, /a/, and /u/, and the stop consonants /p/, /t/, /k/, /b/, /d/, and /g/, as well as the nasals /m/ and /n/ appear in a large number of spoken languages [100]. Another example is that languages are known to systematically restrict the co-occurrence of consonants in their onsets [8]. These restrictions can serve as learning constraints of, for example, the DPHMM model and may help the model to discover more precise phonetic structures. The challenge of this line of research lies in how to effectively encode these universal constraints as a part of the model.

In addition to universal linguistic knowledge, other simple data pre-processing may be used to improve the quality of the models presented in this thesis. For example, as shown in Fig. 2.6, the DPHMM model learns many units for silence, possibly due to the inconsistent behavior of the Dirichlet process mixture models for learning the number of components from data [130]. To prevent the model from learning the silence units, a practical solution could be using a simple phonetic recognizer to detect speech and non-speech segments in the acoustic data before training the DPHMM model.

■ 6.2.3 Application to Other Languages

Most of the models proposed in this thesis do not assume prior linguistic knowledge. However, the design of the word pronunciation learning model presented in Chapter 5 is largely inspired by linguistic properties observed in English. For example, in order to accommodate the pronunciation of the letter *x*, the maximal number of phonetic units that a letter can map to is set to two. In addition, the number of phonetic units to be discovered is limited to 100, which is roughly an upper bound of the size of the English phoneme set. Given the rationale behind the model design, we envision that to apply this model to other languages, the model structure will need to be modified slightly. For instance, to apply the model to tonal languages, by viewing vowels with different tones as distinct phones, we may consider increasing the number of the phonetic units to be discovered in the model.

■ 6.2.4 Semi-supervised Pronunciation Lexicon Learning

Chapter 5 of this thesis suggests that a pure data-driven approach is viable for training ASR systems. However, instead of regarding our approach as a replacement, we view it as a promising augmentation to the current supervised training procedure. For example, rather than create a pronunciation dictionary for every word in a language, we can have the linguistic experts only denote the pronunciations for a small set of frequently-used words in the language, and use this compact lexicon as a starting point for training our model. By seeding our model with some expert knowledge, we can guide the learning of our model better and potentially allow the model to discover more accurate relationship between the written and spoken systems of the language. We believe that a hybrid framework that combines expert-defined and automatically-induced knowledge is the key to successfully developing speech recognizers for many more languages in the world, especially for low-resource languages.

6.2.5 Progressive Training

Compared to child-directed speech, which usually consists of short utterances and abundant repeated words, the speech data that our models are trained on tend to contain longer sentences and a larger vocabulary. This suggests that, compared to infants, our models are learning in a relatively tougher condition. If a simpler subset of the training data for any of our models can be automatically selected, we may be able to ground the learning of the model with this smaller dataset first, and then gradually add in more complicated training samples to grow the model. The advantage of this progressive training strategy is that it provides a strong learning constraint to the model at the early training stage, which can potentially allow the model to discover more accurate latent structures. The major difficulty for implementing this training strategy is how to automate the data selection process.

6.2.6 Learning from More Sensory Data

The research presented in this thesis is mostly limited to speech data. However, during the language acquisition process, human infants are exposed to much more sensory information than simply the speech stream. One obvious example of other sensory inputs is the visual stream. Recent research [81] has applied adaptor grammars to simultaneously segment words from phoneme strings and learn the referents of some of the words, in which the visual stream associated with each utterance (represented as a phoneme string) is mapped as a symbolic representation. While promising results were obtained in [81], the input to the adaptor grammar is once again highly abstracted. Therefore, one interesting question is whether we can develop a system that learns from less processed data, similar to those employed in [158] for grounding language acquisition from acoustic and video input. Can we discover linguistic structures by leveraging information in the visual data? Can the synergies between speech and visual segmentation help learn each other?

6.2.7 An Analysis-by-synthesis Approach to Augmenting Model Design

Far from the final word on one-shot learning of spoken words, we consider our investigation to be a first step towards understanding how adults and children learn novel phonological sequences from just one example. We can envision many ways to further expand the scale of our work. For example, rooted in the principles of compositionality and causality, the authors of [104] take an *analysis-by-synthesis* approach to designing a generative model that achieves a human-level performance on one-shot learning tasks for classifying and generating hand-

written characters. Can the same approach be applied to design a computational model that achieves human-level performance on one-shot learning tasks for spoken words? More specifically, the term of *analysis-by-synthesis* in the speech community often refers to explicit modeling of the articulatory process [65], which requires speech data augmented with articulatory measurements from the speaker's vocal tract [5]. Can this type of information be included as part of a generative model? While we choose to build our model on a feature representation that only implicitly reflects this type of information [9], we see the potential in taking the *analysis-by-synthesis* approach to building models that more closely resemble the way humans produce speech.

Appendix A

Moses Translations of the Discovered Word Pronunciations

In this appendix, we present more Moses translations of the word pronunciations that are discovered by the lexicon induction model described in Chapter 5. As in Table 5.4, each entry of the following table consists of 1) the automatically-induced word pronunciation, 2) the corresponding translation denoted in standard phone units, which is obtained by using the Moses translation system described in 5.5, and 3) the expert-defined pronunciation.

Word	automatically-induced pronunciation	Translated automatically-induced pronunciation	Expert-defined pronunciation
freeze	10 66 28 50 20 82	f r iy z	f r iy z
greenland	19 28 20 80 45 15 46 25 22 81	gcl g r iy n l ax n	gcl g r iy n l ax n dcl d
columbus	19 66 6 88 12 63 9 13 51 24 93	kcl k el ah m bcl b ax s	kcl k ax l ah m bcl b ax s
night	22 75 15 29 59 3 86	n ay tcl t	n ay tcl t
in	25 22	ax n	ax n
rains	28 7 22 93	r ey n epi z	r ey n z
find	31 18 12 46	f ay n	f ay n dcl d
nice	33 58 62 57 24 10	n ay s epi	n ay s
pegasus	35 86 29 59 13 37 10 69 24	pcl p ey gcl g ax s ax s	pcl p eh gcl g ax s ax s
mountain	39 79 45 55 3 94 22 33	m aw n tcl t ax n	m aw n tcl t ax n

Continued on next page

Table A.1 – continued from previous page

Word	automatically-induced pronunciation	Translated automatically-induced pronunciation	Expert-defined pronunciation
way	40 12	w ay	w ey
bay	42 29 73	bcl b ey	bcl b ey
accumulation	5 27 84 80 92 80 92 85 34 22 92	ax kcl k y uw m y ax l ey sh ax n	ax kcl k y uw m y ax l ey sh ax n
yesterday	50 16 89 27 44 27 13 25 29 97	y eh s tcl t er dcl d ey	y eh s tcl t er dcl d ey
yesterday	50 16 93 34 44 27 25 29 97 38	y eh s tcl t er dcl d ey	y eh s tcl t er dcl d ey
accumulation	59 60 84 80 92 80 91 85 34 22 38	ax kcl k y uw m y ax l ey sh en	ax kcl k y uw m y ax l ey sh en
asia	62 57 34 47 49 70	ey zh ax	ey zh ax
vail	83 62 77 79 63	v ey l	v ey l
not	83 75 11 35	n aa tcl	n aa tcl t
heat	84 20 60	hh iy tcl t	hh iy tcl t
hot	86 66 4 65 3	hh aa tcl t	hh aa tcl t
soon	89 94 41 22	s uw n	s uw n
san	93 94 29 77 22	s ae n	s ae n
weather	96 98 74 64 30	w eh dh er	w eh dh er
weather	98 74 64 30 54	w eh dh er	w eh dh er
able	1 21 14	bcl b el	ey bcl b ax l
daylight	1 29 91 76 15 59	dcl d ey l ay kcl	dcl d ey l ay tcl t
marlborough	1 39 56 88 42 67 56 56 88	m aa r l bcl b aa r ow	m aa r l bcl b r ow
seventy	10 26 99 26 62 33 99	s eh v ey n iy	s eh v ax n t tcl iy
flights	10 40 12 59 3 93	f l ay tcl t s	f l ay tcl t s
forecasted	10 6 43 55 3 47 16 24 37 93	f ao r kcl k ae s ax s	f ao r kcl k ae s t tcl ax dcl
freeze	10 66 28 33 57 10 26	f r iy s	f r iy z

Continued on next page

Table A.1 – continued from previous page

Word	automatically-induced pronunciation	Translated automatically-induced pronunciation	Expert-defined pronunciation
davis	13 73 64 51 93	dcl d ey dx ax s	dcl d ey v ax s
mountain	14 83 49 79 63 3 86 83 22 38	m ow n tcl t ax n	m aw n tcl t ax n
couple	19 14 86 35 40 5	kcl k s pcl p ax l	kcl k ah pcl p ax l
precipitation	19 66 27 93 37 35 37 35 47 57 34 25 38	pcl p axr s ax pcl p ax tcl t ey sh ax n	pcl p axr s ih pcl p ax tcl t ey sh ax n
bermuda	19 67 67 9 50 80 73 13 16 70	bcl b er m y uw dcl d iy ax	bcl b er m y uw dcl d ax
need	22 50 20 60	n iy tcl	n iy dcl
flights	24 6 76 59 73 55 10	f l ay tcl t s	f l ay tcl t s
anytime	25 50 20 60 94 15 9 92	eh n iy tcl t ax m	eh n iy tcl t ay m
carolina	27 47 2 77 15 15 4 2 33 58 70	kcl k ae n ax l ay n ax	kcl k ae r ax l ay n ax
asai	29 73 89 94 16 70	ey s ax	ey zh ax
find	31 18 12 46	f ay n	f ay n dcl d
knox	33 15 19 72 72	n aa kcl k	n aa kcl k s
shreveport	34 44 28 95 3 40 43 27 3 81	sh r iy s pcl p ao r tcl t	sh r iy v pcl p ao r tcl t
george	34 80 43 67 27 34	dcl jh y ao r dcl jh	dcl jh ao r dcl jh
plus	35 18 45 89	pcl p ax s	pcl p l ah s
music	39 50 95 69 59 27 72	m y uw n ax kcl k	m y uw z ax kcl k
baton	42 16 35 60 94	bcl b ae tcl t	bcl b ae tcl en
old	43 97 1	ao l dx	ow l dcl d
understand	5 46 27 89 95 25 22	ah n d er s tcl t ax n	ah n d er s tcl t ae n dcl d
done	64 75 9 38	dcl d ax n	dcl d ah n
were	68 98 85	w ih	w er
couple	72 19 79 35 18 23	kcl k ae pcl p el	kcl k ah pcl p ax l

Continued on next page

Table A.1 – continued from previous page

Word	automatically-induced pronunciation	Translated automatically-induced pronunciation	Expert-defined pronunciation
carolina	72 47 17 79 76 12 15 46 75 70	kcl k ae l ay n ax	kcl k ae r l ay n ax
couple	72 86 79 35 6 88	kcl k ae pcl p el	kcl k ah pcl p ax l
las	76 15 49 93 89	l ax s	l aa s
pocatello	81 19 6 23 19 37 86 79 76 63 88 81	ax pcl p ow kcl k ax tcl t eh l ow	pcl p ow kcl k ax tcl t eh l ow
boise	83 42 43 85 62 10 36 97 81	bcl b oy s iy iy	bcl b oy s iy
nebraska	83 91 42 28 17 16 93 72 16 81	m ax bcl b r ae s kcl k ax	n ax bcl b r ae s kcl k ax
marseille	9 39 56 27 93 69 29 73 70	m aa r s ey ax	m aa r s ey
manitoba	9 77 92 35 60 94 23 42 87 70	m ae n tcl t ow bcl b ax	m ae n ax tcl t ow bcl b ax
might	9 83 39 59 20	m ay n	m ay dx
season	93 20 93 94 22	s iy s ax n	s iy z ax n
forty	93 6 43 27 95	f ao r tcl t	f ao r tcl t iy
san	93 94 29 77 22	s ae n	s ae n
wear	96 43 28 71	w ao r	w eh r
wind	96 98 85 25 22	w ih n	w ih n dcl
diego	13 20 73 59 80 49 23 23	dcl d iy gcl g ow l	dcl d iy ey gcl g ow
prague	19 66 56 63 45 19 86	pcl p r aa kcl k	pcl p r aa gcl g

Table A.1: More examples of the automatically inferred word pronunciations and their translations denoted in expert-defined phonetic units, which are generated by using a Moses translation system. The corresponding expert-defined pronunciation for each word is listed in the right-most column.

Bibliography

[1] Speech Signal Processing Toolkit (SPTK), 2013.

[2] Guillaume Aimetti. Modelling early language acquisition skills: Towards a general statistical learning mechanism. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 1–9. Association for Computational Linguistics, 2009.

[3] Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974.

[4] Michiel Bacchiani and Mari Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29:99 – 114, 1999.

[5] Ziad Al Bawab, Bhiksha Raj, and Richard M. Stern. Analysis-by-synthesis features for speech recognition. In *Processings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[6] Issam Bazzi. Modelling out-of-vocabulary words for robust speech recognition. Ph.d. thesis, Massachusetts Institute of Technology, 2002.

[7] Matthew J. Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. The infinite hidden Markov model. *In Advances in Neural Information Processing Systems*, 14:577 – 584, 2002.

[8] Iris Berent and Tracy Lennertz. Universal constraints on the sound structure of language: Phonological or acoustic? *Journal of Experimental Psychology: Human perception and performance*, 36(1):212, 2010.

[9] Thomas G. Bever and David Poeppel. Analysis by synthesis: a (re-)emerging program of research for language and vision. *Biolinguistics*, 4:174–200, 2010.

[10] Maximilian Bisani and Hermann Ney. Open vocabulary speech recognition with flat hybrid models. In *Proceedings of the European Conference on Speech Communication and Technology*, 2005.

[11] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, May 2008.

[12] Benjamin Börschinger and Mark Johnson. Exploring the role of stress in Bayesian word segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 2:93 – 104, 2014.

[13] Michael R. Brent and Timothy A. Cartwright. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1):93–125, 1996.

[14] Chris Callison-Burch and Mark Dredze. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12. Association for Computational Linguistics, 2010.

[15] Susan Carey and Elsa Bartlett. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17 – 29, August 1978.

[16] Chun-an Chan and Lin-shan Lee. Unsupervised spoken-term detection with spoken queries using segment-based dynamic time warping. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 693–696, 2010.

[17] Chun-An Chan and Lin-Shan Lee. Unsupervised hidden Markov modeling of spoken queries for spoken term detection without speech recognition. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2141 – 2144, 2011.

[18] Marie Cheour, Rita Ceponiene, Anne Lehtokoski, Aavo Luuk, Jüri Allik, Kimmo Alho, and Risto Näätänen. Development of language-specific phoneme representations in the infant brain. *Nature neuroscience*, 1(5):351–353, 1998.

[19] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[20] Cheng-Tao Chung, Chun-an Chan, and Lin-shan Lee. Unsupervised discovery of linguistic structure including two-level acoustic patterns using three cascaded stages of iterative optimization. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8081–8085. IEEE, 2013.

[21] Eve V. Clark. *First language acquisition*. Cambridge University Press, 2009.

[22] Matthew J. C. Crump, John V. McDonnell, and Todd M. Gureckis. Evaluating Amazon's Mechanical Turk as a Tool for Experimental Behavioral Research. *PLoS ONE*, 8(3), March 2013.

[23] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[24] Bart de Boer and Patricia K. Kuhl. Investigating the role of infant-directed speech with a computer model. *Acoustics Research Letters Online*, 4(4):129–134, 2003.

[25] Carl G. de Marcken. *Unsupervised Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, 1996.

[26] Qing Dou and Kevin Knight. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1668–1676, 2013.

[27] Mark Dredze, Aren Jansen, Glen Coppersmith, and Ken Church. NLP on spoken documents without ASR. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 460–470. Association for Computational Linguistics, 2010.

[28] Sorin Dusan and Lawrence Rabiner. On the relation between maximum spectral transition positions and phone boundaries. In *Proceedings of the 7th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1317 – 1320, 2006.

[29] James H. Moor (Editor). *The Turing Test: The Elusive Standard of Artificial Intelligence*. Kluwer Academic Publishers, 2004.

[30] Stuart M. Shieber (Editor). *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*. MIT Press, September 2005.

[31] Peter D. Eimas. Auditory and phonetic coding of the cues for speech: Discrimination of the [rl] distinction by young infants. *Perception & Psychophysics*, 18(5):341–347, 1975.

[32] Daniel P. W. Ellis. RASTA/PLP/MFCC feature calculation and inversion, 2005.

[33] Micha Elsner, Sharon Goldwater, and Jacob Eisenstein. Bootstrapping a unified model of lexical and phonetic acquisition. In *Proceedings of the 50th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 184–193, 2012.

[34] Micha Elsner, Sharon Goldwater, Naomi Feldman, and Frank Wood. A joint learning model of word segmentation, lexical acquisition, and phonetic variability. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 42–54, 2013.

[35] Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[36] Katharine Graf Estes, Julia L. Evans, Martha W. Alibali, and Jenny R. Saffran. Can infants map meaning to newly segmented words? Statistical segmentation and word learning. *Psychological Science*, 18(3):254–260, 2007.

[37] Yago Pereiro Estevan, Vincent Wan, and Odette Scharenborg. Finding maximum margin segments in speech. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 937 – 940, 2007.

[38] Li Fe-Fei, Robert Fergus, and Pietro Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the Ninth International Conference on Computer Vision*, pages 1134–1141. IEEE, 2003.

[39] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[40] Naomi H. Feldman, Thomas L. Griffiths, Sharon Goldwater, and James L. Morgan. A role for the developing lexicon in phonetic category acquisition. *Psychological Review*, 120(4):751–778, 2013.

[41] Naomi H. Feldman, Thomas L. Griffiths, and James L. Morgan. Learning phonetic categories by learning a lexicon. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 2208–2213, 2009.

[42] Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[43] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 618–626, 2006.

[44] Jonathan Foote, Steve J. Young, Gareth J. F. Jones, and Karen Spärck Jones. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech & Language*, 11(3):207–224, 1997.

[45] Emily Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. A sticky HDP-HMM with application to speaker diarization. *Annals of Applied Statistics*, 2011.

[46] Emily B. Fox. *Bayesian nonparametric learning of complex dynamical phenomena*. PhD thesis, Massachusetts Institute of Technology, 2009.

[47] Michael C. Frank, Noah D. Goodman, and Joshua B. Tenenbaum. Using speakers' referential intentions to model early cross-situational word learning. *Psychological Science*, 20(5):578–585, 2009.

[48] Dennis Butler Fry. *Homo loquens : Man as a talking animal*. Cambridge University Press, 1977.

[49] Toshiaki Fukada, Michiel Bacchiani, Kuldip Paliwal, and Yoshinori Sagisaka. Speech recognition based on acoustically derived segment units. In *Proceedings of ICSLP*, pages 1077 – 1080, 1996.

[50] Alvin Garcia and Herbert Gish. Keyword spotting of arbitrary words using minimal speech resources. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 949–952, 2006.

[51] John Garofalo, David Graff, Doug Paul, and David Pallett. CSR-I (WSJ0) Other. Linguistic Data Consortium, Philadelphia, 1993.

[52] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallet, Nancy L. Dahlgren, and Victor Zue. Timit acoustic-phonetic continuous speech corpus, 1993.

[53] Jean-Luc. Gauvain, Abdelkhalek Messaoudi, and Holger Schwenk. Language recognition using phone lattices. In *Proceedings of the 5th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1283–1286, 2004.

[54] Hadrien Gelas, Solomon Teferra Abate, Laurent Besacier, and François Pellegrino. Quality assessment of crowdsourcing transcriptions for African languages. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3065–3068, 2011.

[55] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, second edition, 2004.

[56] James Glass. *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition*. Ph. d. thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1988.

[57] James Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17:137 – 152, 2003.

[58] James Glass. Towards unsupervised speech processing. In *Proceedings of International Conference on Information Sciences, Singal Processing and their Applications (ISSPA)*, pages 1–4, 2012.

[59] James Glass, Timothy J. Hazen, Lee Hetherington, and Chao Wang. Analysis and processing of lecture audio data: Preliminary investigations. In *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL*, pages 9–12. Association for Computational Linguistics, 2004.

[60] Sharon Goldwater. A Bayesian framework for word segmentation: exploring the effects of context. *Cognition*, 112:21–54, 2009.

[61] Rebecca L. Gomez and LouAnn Gerken. Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition*, 70(2):109–135, 1999.

[62] Rebecca L Gómez and LouAnn Gerken. Infant artificial language learning and language acquisition. *Trends in cognitive sciences*, 4(5):178–186, 2000.

[63] Joshua T. Goodman. *Parsing Inside-Out*. PhD thesis, Harvard University, 1998.

[64] Jan V Goodsit, James L Morgan, and Patricia K Kuhl. Perceptual strategies in prelingual speech segmentation. *Journal of child language*, 20:229–229, 1993.

[65] Morris Halle and Kenneth N. Stevens. Speech recognition: A model and a program for research. *IRE Transactions on Information Theory*, 8(2):155–159, 1962.

[66] David Frank Harwath. Unsupervised modeling of latent topics and lexical units in speech audio. Master's thesis, Massachusetts Institute of Technology, 2013.

[67] Timothy J. Hazen, Wade Shen, and Christopher White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 421–426. IEEE, 2009.

[68] Lee I. Hetherington. The MIT finite-state transducer toolkit for speech and language processing. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2609–2612, 2004.

[69] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj. Unsupervised word segmentation from noisy input. In *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 458–463. IEEE, 2013.

[70] Marijn Huijbregts, Mitchell McLaren, and David van Leeuwen. Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4436–4439. IEEE, 2011.

[71] Lee S. Hultzén, Joseph H. D. Allen Jr., and Murray S. Miron. *Tables of Transitional Frequencies of English Phonemes*. University of Illinois Press, Urbana, 1964.

[72] Hemant Ishwaran and Mahmoud Zarepour. Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.

[73] Katunobu Itou, Mikio Yamamoto, Kazuya Takeda, Toshiyuki Takezawa, Tatsuo Matsuoka, Tetsunori Kobayashi, and Kiyohiro Shikano. JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research. *Journal of Acoustical Society of Japan*, 20:199–206, 1999.

[74] Aren Jansen and Kenneth Church. Towards unsupervised training of speaker independent acoustic models. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1693 – 1696, 2011.

[75] Aren Jansen, Kenneth Church, and Hynek Hermansky. Towards spoken term discovery at scale with zero resources. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1676–1679, 2010.

[76] Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532 – 556, 1976.

[77] Mark Johnson. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[78] Mark Johnson. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 398–406. The Association for Computer Linguistics, 2008.

[79] Mark Johnson. Pitman-Yor adaptor grammar sampler, 2013.

[80] Mark Johnson and Katherine Demuth. Unsupervised phonemic chinese word segmentation using adaptor grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 528–536, August 2010.

[81] Mark Johnson, Katherine Demuth, Bevan K. Jones, and Michael J. Black. Synergies in learning words and their referents. *In Advances in Neural Information Processing Systems*, pages 1018–1026, 2010.

[82] Mark Johnson and Sharon Goldwater. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 317–325, 2009.

[83] Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. *In Advances in Neural Information Processing Systems*, pages 641–648, 2006.

[84] Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of Human Language Technologies: The 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 139–146, 2007.

[85] Matthew J. Johnson and Alan S. Willsky. Bayesian nonparametric hidden semi-Markov models. *Journal of Machine Learning Research*, 14:673–701, February 2013.

[86] Gareth J. F. Jones, Jonathan T. Foote, Karen Spärck Jones, and Steve J. Young. Retrieving spoken documents by combining multiple index sources. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 30–38, New York, NY, USA, 1996. ACM.

[87] Biing Hwang Juang and Laurence Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[88] Peter W. Jusczyk and Paul A Luce. Infants' sensitivity to phonotactic patterns in the native language. *Journal of Memory and Language*, 33(5):630–645, 1994.

[89] Sadik Kapadia, Valtcho Valtchev, and Steve Young. MMI training for continuous phoneme recognition on the TIMIT database. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 491–494, 1993.

[90] Sawit Kasuriya, Virach Sornlertlamvanich, Patcharika Cotsomrong, Supphanat Kanokphara, and Nattanun Thatphithakkul. Thai speech corpus for Thai speech recognition. In *Proceedings of Oriental COCOSDA*, pages 54–61, 2003.

[91] Mirjam Killer, Sebastian Stüker, and Tanja Schultz. Grapheme based speech recognition. In *Proceeding of the Eurospeech*, pages 3141–3144, 2003.

[92] Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. Unsupervised analysis for decipherment problems. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 499–506, 2006.

[93] Kevin Knight and Kenji Yamada. A computational approach to deciphering unknown scripts. In *ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 37–44, 1999.

[94] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[95] Henry. Kucera and W. Nelson Francis. *Computational analysis of present-day American English*. Brown University Press, Providence, RI, 1967.

[96] Patricia K. Kuhl. Early linguistic experience and phonetic perception: Implications for theories of developmental speech perception. *Journal of Phonetics*, 1993.

[97] Patricia K. Kuhl. Early language acquisition: Cracking the speech code. *Nature reviews neuroscience*, 5(11):831–843, 2004.

[98] Patricia K. Kuhl, Erica Stevens, Akiko Hayashi, Toshisada Deguchi, Shigeru Kiritani, and Paul Iverson. Infants show a facilitation effect for native language phonetic perception between 6 and 12 months. *Developmental science*, 9(2):F13 – F21, 2006.

[99] Patricia K. Kuhl, Karen A. Williams, Francisco Lacerda, Kenneth N. Stevens, and Björn Lindblom. Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, 255(5044):606–608, 1992.

[100] Peter Ladefoged and Ian Maddleson. *The Sounds of the World's Languages*. Wiley-Blackwell, first edition, 1996.

[101] Brenden M. Lake, Chia-ying Lee, James R. Glass, and Joshua B. Tenenbaum. One-shot learning of generative speech concepts. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Soceity*, 2014.

[102] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 2568–2573, 2011.

[103] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Concept learning as motor program induction: A large-scale empirical study. In *Proceedings of the 34rd Annual Conference of the Cognitive Science Society*, pages 659 – 664, 2012.

[104] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems*, 2013.

[105] Lori F. Lamel and Jean-Luc Gauvain. Language identification using phone-based acoustic likelihoods. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 293 – 296. IEEE, 1994.

[106] Lori F. Lamel, Robert H. Kassel, and Stephanie Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 100–110, 1986.

[107] Ian Lane, Alex Waibel, Matthias Eck, and Kay Rottmann. Tools for collecting speech corpora via Mechanical-Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 184–187. Association for Computational Linguistics, 2010.

[108] Jill Lany and Jenny R. Saffran. From statistics to meaning: Infants' acquisition of lexical categories. *Psychological Science*, 2010.

[109] Karim Lari and Steve J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56, 1990.

[110] Chia-ying Lee and James Glass. A transcription task for crowdsourcing with automatic quality control. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3041–3044, 2011.

[111] Chia-ying Lee and James Glass. A nonparametric Bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 40–49, 2012.

[112] Chia-ying Lee, Yu Zhang, and James Glass. Joint learning of phonetic units and word pronunciations for ASR. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 182–192, 2013.

[113] Chin-Hui Lee, Frank Soong, and Biing-Hwang Juang. A segment model based approach to speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 501–504, 1988.

[114] Hung-yi Lee, Yun-Chiao Li, Cheng-Tao Chung, and Lin-shan Lee. Enhancing query expansion for semantic retrieval of spoken content with automatically discovered acoustic patterns. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8297–8301. IEEE, 2013.

[115] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1641 – 1648, 1989.

[116] Yun-Chiao Li, Hung-yi Lee, Cheng-Tao Chung, Chun-an Chan, and Lin-shan Lee. Towards unsupervised semantic retrieval of spoken content with query expansion based on automatically discovered acoustic patterns. In *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 198–203. IEEE, 2013.

[117] Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 688–697, 2007.

[118] Alvin M. Liberman, Franklin S. Cooper, Donald P. Shankweiler, and Michael Studdert-Kennedy. Perception of the speech code. *Psychological review*, 74(6):431, 1967.

[119] Constantine Lignos. Modeling infant word segmentation. In *Proceedings of the 15th Conference on Computational Natural Language Learning*, pages 29–38. Association for Computational Linguistics, 2011.

[120] Steven N. MacEachern and Peter Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

[121] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. Vocabulary independent spoken term detection. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 615–622. ACM, 2007.

[122] Anupam Mandal, K. R. Prasanna Kumar, and Pabitra Mitra. Recent developments in spoken term detection: a survey. *International Journal of Speech Technology*, pages 1–16, 2013.

[123] Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky. Using the Amazon Mechanical Turk for transcription of spoken language. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5270–5273. IEEE, 2010.

[124] Jessica Maye, Janet F. Werker, and LouAnn Gerken. Infant sensitivity to distributional information can affect phonetic discrimination. *Cognition*, 82(3):B101–B111, 2002.

[125] Ian McGraw, Ibrahim Badr, and James Glass. Learning lexicons from speech using a pronunciation mixture model. *IEEE Transactions on Speech and Audio Processing*, 21(2):357–366, 2013.

[126] Ian McGraw, Chia-ying Lee, Lee Hetherington, Stephanie Seneff, and Jim Glass. Collecting voices from the cloud. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 1576 – 1583, 2010.

[127] Fergus R. McInnes and Sharon Goldwater. Unsupervised extraction of recurring words from infant-directed speech. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 2006 – 2011, 2011.

[128] Jacques Mehler and Robin (Editors) Fox. *Neonate cognition: Beyond the blooming buzzing confusion*. Lawrence Erlbaum Associates, 1985.

[129] David R. H. Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, and Herbert Gish. Rapid and accurate spoken term detection. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 314–317, 2007.

[130] Jeffrey W. Miller and Matthew T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. *Advances in Neural Information Processing Systems*, pages 199–206, 2013.

[131] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint*

Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (AFNLP), pages 100–108. Association for Computational Linguistics, 2009.

[132] Kevin P. Murphy. Hidden semi-Markov models (hsmms). Technical report, University of British Columbia, 2002.

[133] Kevin P. Murphy. Hidden semi-Markov models (segment models). Technical report, November 2002.

[134] Kevin P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia, 2007.

[135] Cory S. Myers and Lawrence R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):284–297, 1981.

[136] Radford M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[137] Graham Neubig, Masato Mimura, and Tatsuya Kawahara. Bayesian learning of a language model from continuous speech. *The Institute of Electronics, Information and Communication Engineers (IEICE) TRANSACTIONS on Information and Systems*, 95(2):614–625, 2012.

[138] Hermann Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):263–271, 1984.

[139] Atta Norouzian, Richard Rose, Sina Hamidi Ghalehjegh, and Aren Jansen. Zero resource graph-based confidence estimation for open vocabulary spoken term detection. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8292–8296. IEEE, 2013.

[140] Timothy John O'Donnell. *Productivity and reuse in language*. PhD thesis, Harvard University, 2011.

[141] Kota Ohata. Phonological differences between Japanese and English: Several Potentially Problematic Areas of Pronunciation for Japanese ESL/EFL Learners. *Asian EFL Journal*, 6(4):1–19, 2004.

[142] Hideo Okada. Japanese. *Journal of the International Phonetic Association*, 21(2):94 – 96, 1991.

[143] Timothy J. O'donnell, Jesse Snedeker, Joshua B. Tenenbaum, and Noah D. Goodman. Productivity and reuse in language. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 1613 – 1618, 2011.

[144] Kuldip Paliwal. Lexicon-building methods for an acoustic sub-word based speech recognizer. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 729–732, 1990.

[145] Carolina Parada, Mark Dredze, Abhinav Sethy, and Ariya Rastrow. Learning sub-word units for open vocabulary speech recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 712–721. Association for Computational Linguistics, 2011.

[146] Alex S. Park and James R. Glass. Unsupervised pattern discovery in speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 16(1):186–197, 2008.

[147] Slav Petrov, Adam Pauls, and Dan Klein. Learning structured models for phone recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 897 – 905, 2007.

[148] Steven Pinker. *The language instinct*. William Morrow & Company, 1994.

[149] Jim Pitman. Combinatorial stochastic processes. Technical report, 621, Department of Statistics, UC Berkeley, 2002. Lecture notes for St. Flour course, 2002.

[150] Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

[151] Kristen Precoda. Non-mainstream languages and speech recognition: Some challenges. *Journal of Computer-Assisted Language Instruction Consortium*, 21(2):229–243, 2004.

[152] Yu Qiao, Naoya Shimomura, and Nobuaki Minematsu. Unsupervised optimal phoeme segmentation: Objectives, algorithms and comparisons. In *Proceedings of International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3989 – 3992, 2008.

[153] Lawrence Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[154] Carl Edward Rasmussen. The infinite Gaussian mixture model. *In Advances in Neural Information Processing Systems*, 12:554–560, 2000.

[155] Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran. A new method for OOV detection using hybrid word/fragment system. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3953–3956, 2009.

[156] Ariya Rastrow, Abhinav Sethy, Bhuvana Ramabhadran, and Frederick Jelinek. Towards using hybrid word and fragment units for vocabulary independent LVCSR systems. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1931–1934, 2009.

[157] Sujith Ravi and Kevin Knight. Bayesian inference for Zodiac and other homophonic ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 239–247, 2011.

[158] Deb Roy. Grounded spoken language acquisition: Experiments in word learning. *IEEE Transactions on Multimedia*, 5(2):197–209, 2003.

[159] Jenny R. Saffran. Constraints on statistical language learning. *Journal of Memory and Language*, 47(1):172–196, 2002.

[160] Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

[161] Tara N. Sainath, Bhuvana Ramabhadran, and Michael Picheny. An exploration of large vocabulary tools for small vocabulary phonetic recognition. In *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 359–364. IEEE, 2009.

[162] Hiroaki Sakoe. Two-level DP-matching – a dynamic programming-based pattern matching algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(6):588–595, 1979.

[163] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[164] Ruslan Salakhutdinov, Joshua B. Tenenbaum, and Antonio Torralba. Learning with hierarchical-deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1958–1971, 2013.

[165] Odette Scharenborg, Vincent Wan, and Mirjam Ernestus. Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries. *Journal of the Acoustical Society of America*, 127:1084–1095, 2010.

[166] Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[167] Fei Sha and Lawrence K. Saul. Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 313–316, 2007.

[168] Harvey F. Silverman and David P. Morgan. The application of dynamic programming to connected speech recognition. *IEEE Acoustics, Speech, and Signal Processing (ASSP) Magazine*, 7(3):6–25, 1990.

[169] Man-hung Siu, Herbert Gish, Arthur Chan, William Belfield, and Steve Lowe. Unsupervised training of an HMM-based self-organizing unit recgonizer with applications to topic classification and keyword discovery. *Computer, Speech, and Language*, 2013.

[170] Linda Smith and Chen Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106(3):1558–1568, 2008.

[171] Benjamin Snyder, Regina Barzilay, and Kevin Knight. A statistical model for lost language decipherment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1048–1057. Association for Computational Linguistics, 2010.

[172] Kenneth N. Stevens. *Acoustic Phonetics*. MIT Press, 2000.

[173] Sebastian Stüker and Tanja Schultz. A grapheme based speech recognition system for Russian. In *Proceedings of the 9th Conference Speech and Computer*, 2004.

[174] Daniel Swingley. Statistical clustering and the contents of the infant vocabulary. *Cognitive psychology*, 50(1):86–132, 2005.

[175] Zheng-Hua Tan and Børge Lindberg. High-accuracy, low-complexity voice activity detection based on a posteriori SNR weighted energy. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2231–2234, 2009.

[176] Zheng-Hua Tan and Børge Lindberg. Low-complexity variable frame rate analysis for speech recognition and voice activity detection. *IEEE Journal of Selected Topics in Signal Processing*, 4(5):798–807, 2010.

[177] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2004.

[178] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. Speech synthesis based on Hidden Markov Models. *Proceedings of the IEEE*, 101(5):1234–1252, 2013.

[179] Michael Tomasello. *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press, 2009.

[180] Amir Hossein Harati Nejad Torbati, Joseph Picone, and Marc Sobel. Speech acoustic unit segmentation using hierarchical dirichlet processes. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 637– 641, 2013.

[181] Gautam K. Vallabha, James L. McClelland, Ferran Pons, Janet F. Werker, and Shigeaki Amano. Unsupervised learning of vowel categories from infant-directed speech. *Proceedings of the National Academy of Science*, 104(33):13273–13278, 2007.

[182] Balakrishnan Varadarajan, Sanjeev Khudanpur, and Emmanuel Dupoux. Unsupervised learning of acoustic sub-word units. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 165–168, 2008.

[183] Roy G. Wallace, Robert J. Vogt, and Sridha Sridharan. A phonetic search approach to the 2006 NIST spoken term detection evaluation. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2385–2388, 2007.

[184] Janet F. Werker and Richard C. Tees. Cross-language speech perception: Evidence for perceptual reorganization during the first year of life. *Infant behavior and development*, 7(1):49–63, 1984.

[185] Di Wu, Fan Zhu, and Ling Shao. One shot learning gesture recognition from rgbd images. In *In Computer Vision and Pattern Recognition Workshops*, pages 7–12. IEEE, 2012.

[186] Fei Xu and Joshua B. Tenenbaum. Word learning as Bayesian inference. *Psychological Review*, 114(2):245–272, 2007.

[187] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Duration modeling for HMM-based speech synthesis. In *ICSLP*, volume 98, pages 29–31, 1998.

[188] Steve J. Young, J.J. Odell, and Philip C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of HLT*, pages 307–312, 1994.

[189] Heiga Zen, Keiichi Tokuda, and Alan W. Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

[190] Yaodong Zhang. *Unsupervised speech processing with applications to query-by-example spoken term detection*. PhD thesis, Massachusetts Institute of Technology, 2013.

[191] Yaodong Zhang and James Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 398 – 403, 2009.

[192] Yaodong Zhang, Ruslan Salakhutdinov, Hung-An Chang, and James Glass. Resource configurable spoken query detection using deep Boltzmann machines. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5161–5164, 2012.

[193] Victor Zue, James Glass, Michael Phillips, and Stephanie Seneff. The MIT SUMMIT speech recognition system: A progress report. In *Proceedings of the workshop on Speech and Natural Language*, pages 179–189. Association for Computational Linguistics, 1989.

[194] Victor Zue, Stephanie Seneff, and James Glass. Speech database development at MIT: TIMIT and beyond. *Speech Communication*, 9(4):351 – 356, 1990.

[195] Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington. Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8:85–96, 2000.