Using Graphone Models in Automatic Speech Recognition

by

Stanley Xinlei Wang

S.B. Massachusetts Institute of Technology (2008)

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2009

Certified by .....
James R. Glass
Principal Research Scientist
Thesis Supervisor

Certified by .....
I. Lee Hetherington
Research Scientist
Thesis Co-Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Using Graphone Models in Automatic Speech Recognition

by

Stanley Xinlei Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This research explores applications of joint letter-phoneme subwords, known as graphones, in several domains to enable detection and recognition of previously unknown words. For these experiments, graphones models are integrated into the SUMMIT speech recognition framework. First, graphones are applied to automatically generate pronunciations of restaurant names for a speech recognizer. Word recognition evaluations show that graphones are effective for generating pronunciations for these words. Next, a graphone hybrid recognizer is built and tested for searching song lyrics by voice, as well as transcribing spoken lectures in a open vocabulary scenario. These experiments demonstrate significant improvement over traditional word-only speech recognizers. Modifications to the flat hybrid model such as reducing the graphone set size are also considered. Finally, a hierarchical hybrid model is built and compared with the flat hybrid model on the lecture transcription task.

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

Thesis Co-Supervisor: I. Lee Hetherington
Title: Research Scientist

Acknowledgments

I am grateful to my advisor Jim Glass for his guidance and support. His patience and encouragement has helped me significantly in my past two years at MIT. I would also like to thank my co-advisor, Lee Hetherington, for his guidance and for helping me with building speech recognition components using the MIT FST Toolkit. I am especially appreciative of being able to come by Jim and Lee's offices almost anytime when I have questions, and get them answered very quickly.

Much thanks to Ghinwa Choueiter for teaching me when I first joined the group, and for helping me with the word recognition and lyrics experiments. I would like to thank Paul Hsu for discussions on NLP topics and effective algorithm implementations, Hung-An Chang for help with the lecture transcription experiment, Stephanie Seneff for discussions on subword modeling, and Scott Cyphers for assistance on getting my scripts running. Furthermore, I would like to thank Mark Adler and Jamey Hicks at Nokia Research Cambridge for giving me the opportunity to perform speech research during a summer internship, as well as Imre Kiss, Joseph Polifroni, Jenine Turner, and Tao Wu for their help during my internship.

I am thankful to my friends at MIT for keeping me motivated. Finally, I would like to thank my parents for their continued encouragement and support.

This research was sponsored by Nokia as part of a joint MIT-Nokia research collaboration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Current speech recognition platforms are limited by their knowledge of words and how they are pronounced. Almost all speech recognizers have a pronunciation dictionary that has a finite vocabulary. If the recognizer encounters a word that is not in the speech recognizer's vocabulary, then the recognizer cannot produce a transcription of the word. This limitation is detrimental in many speech recognition applications. For example, if the user wants to find the name of an obscure restaurant while driving home from work, the user could use a spoken dialogue interface in the car, but the dialogue interface may have no idea how the restaurant name should be pronounced, resulting in the inability to understand the user's request and the failure to retrieve information about the restaurant. Another example is transcribing video to enable conventional text-based search. Uncommon words in these videos are often not in the speech recognizer's vocabulary. In addition, because a recognition system needs to find boundaries between words in the input audio stream, a misrecognized word can easily cause the surrounding words, or even the entire sentence to be completely misrecognized. This often results in user frustration and reduced user confidence in speech dialogue systems in general. Therefore, in order to build speech dialogue systems that are applicable in practical user scenarios, we must develop techniques to mitigate the effects of out-of-vocabulary (OOV) words.

There are several methods to combat the OOV problem. One possibility is to dynamically generate pronunciations of OOV words prior to recognition. For a search-by-voice application, pronunciations can be automatically generated for new words as they are imported into a database. These newly generated pronunciations can then be added to the speech recognizer's vocabulary such that they are no longer out-of-vocabulary. However, we may not be able to perform all large vocabulary search-by-voice tasks by increasing the vocabulary. If we increase the vocabulary by large factors, we may be hindered by sparsity in the available language model training data, which could lead to more confusion and degraded recognition performance. Another argument concerns the usability of these search-by-voice interfaces. When a user makes a voice query, if there are truly no matching results, it's better to be able to tell the user that the query produces no results, rather than automatically constraining the audio query to search terms that do produce results, but are not even close to what the user intended. In the latter case, the user may be conviced that the lack of results is because of misrecognition.

Simply increasing the vocabulary of a recognizer also cannot solve the OOV problem in real-time transcription tasks, such as converting streaming broadcast news to text. In this case, we have no way of knowing what the speaker will say, so we have to be able to automatically hypothesize spellings of new words during recognition. Traditional speech recognizers cannot do this because their recognition frameworks are constrained by their known vocabulary, and are only able to output words in the known vocabulary. One may wonder if we could just use a large lexicon as the speech recognizer's vocabulary, and not worry about recognizing the rare esoteric terms in speech. However, in most applications of real-time audio transcription, this approach has a crucial weakness: OOV words are likely to be content words that are highly important for the topic discussed in the audio. In a broadcast news scenario, since newly-coined words frequently appear in headlines, a recognizer with finite vocabulary is likely to miss key words in these news reports.

Vocabulary size growth in training corpora has been characterized in the past [25]. Figure 1-1 shows the the growth of vocabulary size as a function of words in

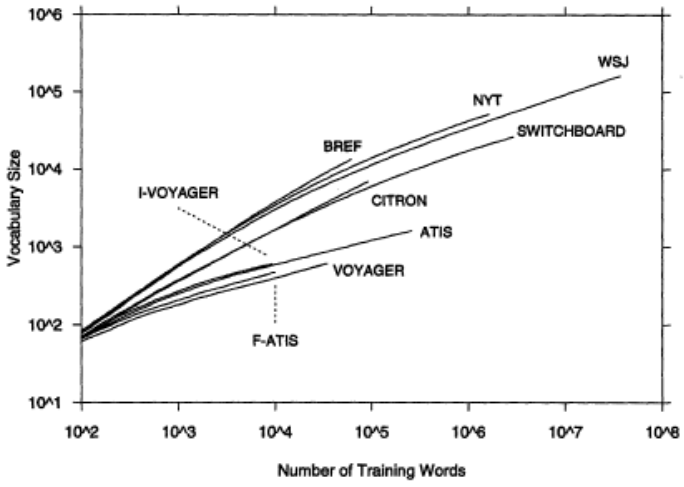

Figure 1-1: Vocabulary sizes of various corpora as more training data is added (from [25]).

several training corpora. These measurements are calculated by first randomizing each corpus and then computing the vocabulary size while scanning the corpus from beginning to end. This process is repeated several times, and the results are averaged to yield the final measurements. ATIS, F-ATIS, VOYAGER are data of users talking with a spoken language system. WSJ, NYT, and BREF are from newspaper text. CITRON and SWITCHBOARD are from human-to-humman speech. In all of these corpora, we can see that even at very large training corpora sizes, the vocabulary size continues to increase rapidly without leveling off. This demonstrates that new words continue to appear even as the training data gets very large. The same study also examines the rate at which new words appear for the same set of corpora, as shown in Figure 1-2. This data shows that as we increase vocabulary coverage, we get still get a substantial amount of new words even at very large vocabulary sizes.

One method for mitigating the effect of OOVs is to use sub-lexical units to model OOV words. These units represent pieces of words, rather than whole words. One of the most successful types of these units are known as *graphones*, which are joint sequences of letters and phonemes [6]. These units are automatically-learned from a pronunciation dictionary so they have potential for applications in any language as long as the written form and pronunciations can be represented as two streams

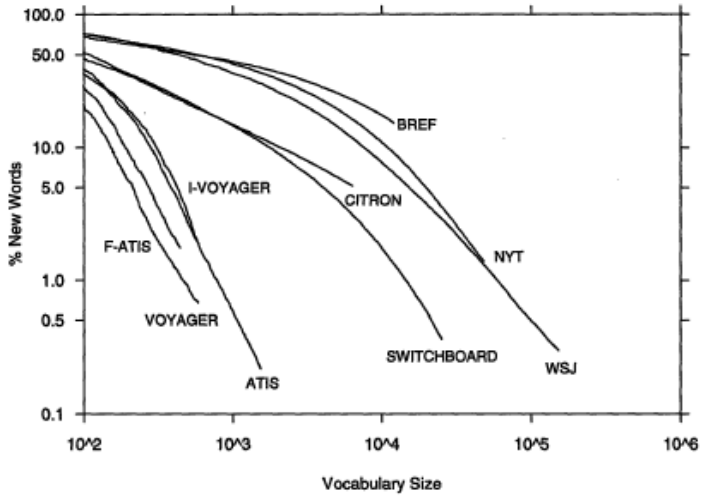

Figure 1-2: New-word rate as the vocabulary size is increased for various corpora (from [25]).

symbols. These units are useful for both automatically generating the pronunciation of words in a dictionary as well as constructing a recognizer that can automatically hypothesize spellings of OOV words. Because of the urgent need for effective OOV models and the success of graphone models so far, we focus the work in this thesis on exploring applications of graphone models in the existing speech recognition framework at MIT.

1.2 Summary of Results

This section contains an overview of the experiments presented in this thesis, along with important results.

- *Generating pronunciations for restaurant and street names.* A graphone-based letter-to-sound converter is built and used to automatically generate pronunciations for restaurant and street names. Word recognition is then used to evaluate the quality of these pronunciations. These experiments show that using a 6-gram, with graphone sizes of 0-1 letters/phonemes, produces the best results. The generalizability of these results are demonstrated through dividing the test set into seen and unseen words. We also explore using multiple L2S

pronunciations for each word, and conclude that using 10 pronunciations per word produces the best performance without increasing confusion within the recognizer. Graphone results are also shown to compare favorably with those of linguistically-motivated subword models.

- *Music lyrics hybrid recognition.* A flat hybrid recognizer is built using graphones, and tested on transcribing spoken lyrics in a closed-vocabulary scenario. The hybrid recognizer significantly outperforms the word-only recognizer at all vocabulary coverage levels. We find that a hybrid recognizer with $\geq$92% vocabulary coverage on the training set actually yield lower recognition error rates than a word-only recognizer with 100% coverage on the training set. The results also demonstrate that graphones are useful for preventing misrecognition of the neighbors of OOV words, as well as hypothesizing the spelling of OOV words. In addition, we experiment with reducing the graphone set significantly, and show that the hybrid recognition performance degrades slightly, but is still significantly better than the word-only recognizer at the same vocabulary coverage level.

- *Transcribing spoken lectures.* Spoken lectures are transcribed using a flat hybrid recognizer at several vocabulary levels. This task is open vocabulary because we cannot know all the words the speaker will say during the lecture. The hybrid recognizer's performance is significantly better than the word-only recognizer at 88-96% vocabulary coverage on the training set, and slightly better than the word-only recognizer at 98-99% coverage levels. We show that the hybrid model's performance remains about the same when the graphone set size is reduced from 19k to 4k. A hybrid recognizer with 100% vocabulary coverage is also built and shown to perform about the same as a word-only recognizer at 100%, even though the graphones can add more confusion to the recognizer. Finally a hierarchical hybrid model is built and compared with the flat hybrid model. Results show that at 98% training set coverage, these two hybrid models perform about the same, and both outperform the word-only recognizer.

1.3 Outline

In the rest of this thesis, we first give an overview of previous works that examine the OOV problem. We then discuss the background technology used in this thesis, which are graphone models and the SUMMIT speech recognition framework. In subsequent chapters, we explore applications of graphone models for speech recognition in areas described in Section 1.2. Finally, we discuss future work that builds upon the research presented in this thesis.

Chapter 2

Background

Since OOV words are inevitable in large vocabulary speech recognition tasks, we must find ways to cope with them in order to build practical speech applications. In this chapter, we first review past techniques for mitigating the effect of out-of-vocabulary (OOV) words on speech recognition performance. Next, we introduce graphone models for both automatic pronunciation generation and hybrid speech recognition. Finally, we give an overview of the SUMMIT speech recognition framework used in this thesis.

2.1 Working with OOVs

Several techniques have been developed to mitigate the effect of OOV words on speech recognition performance. The most straight-forward approach is to increase the vocabulary of the recognizer, but this cannot completely eliminate OOVs in open vocabulary tasks such as audio transcription. Other approaches focus on adding probabilistic machinery to the speech recognizer to detect OOVs in utterances, and possibly even hypothesize pronunciations and spellings associated with the OOV word. These techniques include confidence scoring, filler models, or a combination of both. Multi-stage recognition can also be used to improve large vocabulary recognition by using different recognition strategies in each stage.

2.1.1 Vocabulary Selection and Augmentation

One way to combat OOVs in speech recognition is to select the best set of words to be in-vocabulary for the target domain. If the system has a pretty good idea of what the user might say, the system can be trained with a set of words that give a small OOV rate. One example is a system that provides weather information, which can be trained with a set of geography-related words based on frequency of usage [43]. Martins et al. combine morphological and syntactic information to optimize the trade-off between the predicted vocabulary coverage and the number of added words [30]. Morphological analysis is used to label words with part-of-speech tags, which are then used to optimize vocabulary selection. Another recent study explores using the internet as a source for determining which words to include in the speech recognizer's vocabulary [32]. This technique uses a two-pass strategy for vocabulary selection, where the first pass produces search engine requests, and the second pass augments the speech recognizer's lexicon with words from returned online documents.

We can also increase the vocabulary by automatically generating pronunciations for unknown words. This can be performed by encoding linguistic knowledge, or automatically learning letter-sound associations. The older techniques include purely rule-based approaches that use context-dependent replacement rules [18]. Subsequent techniques have been more data-driven, such as using finite-state transducers [38], or decision trees to capture the mapping between graphemes and phonemes [15]. Another technique is hidden Markov models (HMM), which formulates a generative model for finding the most likely phoneme sequence for generating the observed grapheme sequence [37]. There are discriminative techniques as well, such as perceptron HMM trained in combination with the margin infused relaxed algorithm (MIRA) [27]. Finally, there has also been a series of successful techniques based on sub-lexical models, which are the most relevant to this thesis.

Probabilistic sub-lexical models capture probabilities over chunks of letters commonly referred to as subwords, or sub-lexical units. These units are often correlated to syllables that commonly occur in a language. Some of these models automatically

learn the sub-lexical units, while others have a predefined set of them. However, they all use a probabilistic framework to capture the dependencies between the subwords that make up words in the training set. The advantage of probabilistic approaches is that these methods can generally handle the problem of aligning the grapheme and phoneme sequence much better than many non-probabilistic approaches.

One technique is to use context-free grammar (CFG) parse trees with a superimposed N-gram model to parse words into lexical subwords [31, 34, 35]. First, the pronunciation of words in the lexicon are converted to sequences of subword pronunciations through simple rewrite rules. Next, a rule-based CFG is applied to convert the words in the lexicon to subwords, using the subword pronunciation sequences as a constraint. This process converts the lexicon into lexical subword units containing both letter and sound information, which are also called *spellnemes*. Failing parses are either manually corrected, or made possible by the addition of new rules. This correction process is repeated iteratively until the entire lexicon is successfully converted into subword units. Finally, N-gram probabilities are learned from lexical subword sequences such that this statistical model can be used to segment unknown words into lexical subwords.

Another method is maximum-entropy N-gram modeling of sub-lexical units, in the form of either a conditional model, or a joint model [11]. The conditional model captures probabilities of phonemes conditioned on previous phonemes and letters, over a sliding window. Probability distributions are estimated using maximum entropy with a Gaussian prior. For the joint model, a maximum entropy N-gram model is calculated on training data of words segmented into chunks consisting of either a single letter, a single phoneme, or a single letter-phoneme pair.

Joint grapheme-and-phoneme sequences have been used to model the letter-to-sound relationships. These models, known as *graphone* models, are automatically learned from a pronunciation dictionary through EM training with language modeling. They are described in detail in Section 2.2.1.

It is still an open question whether having a huge vocabulary recognizer is better or worse than an open-vocabulary recognizer that can recognize arbitrary words. One

recent study focused on Italian, and shows that augmenting a speech recognizer's lexicon with automatically-generated pronunciations of OOV words performs slightly better than OOV modeling in an open-vocabulary recognizer [20]. However, in many applications, such as transcribing audio files, there is no way to eliminate all OOV words because we do not know what the speaker will say. In these cases, we can use methods to detect and hypothesize spellings of OOV words.

2.1.2 Confidence Scoring

Confidence scoring allows us to get a sense of how likely a hypothesized word or word sequence is an OOV word. Several criteria can be used to generate the confidence score for region in the recognition hypothesis. They include word posterior probability, acoustic stability, and hypothesis density [41, 28]. Word posterior probability can be computed on the output lattice or the output N-best list of a speech recognizer. Word lattice posterior probabilities are probabilities associated with edges in the word lattice, and can be computed by the well-known forward-backward algorithm. Word N-best list posteriors are different from lattice posteriors in that the N-best list does not have a time associated with each word in the hypothesis. This could be an advantage because the probabilities for the same speech recognizer output are essentially collapsed together. Acoustic stability is a measure of how likely a given word occurs in the same position in all hypothesized N-best list. Hypothesis density is a measure of how many hypotheses have similar likelihoods at a given point in time. The higher the hypothesis density, the more likely the error. Comparison of these measures show that using word posteriors on word lattices yield the best results [41]. Multiple confidence measures can also be combined to construct confidence vectors, which are then passed through a linear classifier to determine if the corresponding word is OOV [24].

More recent studies have explored comparing information from several different sources to detect OOVs. Lin et al. jointly align word and phone lattices to find OOV words [29]. Another study compares confidence measures from strongly constrained and weakly-constrained speech recognizers [10]. The strongly-constrained recognizer

is a regular speech recognizer, while the weakly constrained data comes from a bigram phoneme recognizer and a neural network phone posterior estimator. Word-to-phone transductions have also been combined with phone-lattices to find OOVs [42]. In addition to comparing output of weakly and fully constrained phoneme recognition outputs, this study uses a transducer to decode the a frame-based recognizer's top phoneme output into words, and then compares this output with the word output of a frame-based speech recognizer.

2.1.3 Filler Models

Filler models vary in degrees of sophistication, depending on the application. Some filler models are based on phoneme sequences, and can hypothesize an OOV's pronunciation. Other models are based on joint letter-sound subwords, which not only can hypothesize pronunciation, but also can hypothesize spelling of OOV words. Filler models can be incorporated into a traditional speech recognition network through two main approaches: hierarchical hybrid and flat hybrid. In the hierarchical approach, an OOV tag is added to the speech recognizer's vocabulary. At a given point in decoding an utterance, the recognizer has a choice of decoding the next portion of the utterance as a word, or the OOV tag. This OOV tag is modeled by a separate OOV model that has probabilistic information of phones, syllables, or both. In these recognizers, an extra cost can be added for entering an OOV model because we would prefer to recognize words if the words are in-vocabulary. In the flat-hybrid approach, OOV words in the recognizer's training corpus are turned into subwords or phonemes, which are treated just like regular words in the rest of the recognizer training process.

Bazzi and Glass use phones and syllables to model OOV words through a hierarchical approach [3, 4]. The phoneme-level OOV model is an N-gram model computed on a training corpus of phoneme sequences. Later work in this area use automatically-learned variable-length phonetic units for the OOV model [5]. In this study, mutual information is used as a criterion for merging phonemes into bigger sequences. The authors show that using variable-length units for the OOV model leads to improved recognition performance over using only individual phonemes. They also show that if

individual phonemes were used, it's better to train the OOV model on phoneme sequences in a large-vocabulary dictionary rather than on phoneme sequences of words in a recognizer's LM training corpus. This reduces both domain-dependent bias of the OOV model, as well as bias toward phonemes of frequent words in the LM training corpus.

Modeling OOV words with phonetic units can be effective for OOV detection, but does not allow us to hypothesize the spelling of an OOV word. To do so, our OOV model must incorporate spelling information into subword units. For example, spellnemes can do this by encoding both phonetic and spelling information in each spellneme [35]. In a previous study, Choueiter incorporates spellnemes into a recognizer in a flat-hybrid configuration to model OOV words [12], and demonstrates that using spellnemes in flat hybrid recognition decreases word error rate and sentence error rate for music lyrics spoken by a user with the intention to search for a song. Some OOV words are spelled correctly through spellneme modelling.

Graphones have also been used in flat hybrid recognition [8, 19]. Graphones also contain both spelling and pronunciation information, and are trained in an unsupervised fashion. The details of graphone flat hybrid recognition are described in Section 2.2.2.

2.1.4 Multi-stage Combined Strategies

Efforts have been made to combine confidence scoring and filler model approaches in a multi-stage framework. In the first stage, an utterance is first recognized into subword or phoneme lattices. In the second stage, these lattices are used to generate confidence scores for words or time frames. One technique is to use a phoneme OOV model in the first stage to detect OOV words, and then use confidence scores on remaining words to detect words with confidence [23]. Since filler models can introduce confusion with in-vocabulary words, it would be helpful to have good methods to prevent in-vocabulary words from being detected as OOV words. As an alternative to having a cost for diving into the OOV model, confusion network and vector space model post processing can be used to detect OOVs while accounting for confusion between in-

vocabulary terms and OOV words [33]. Multi-stage frameworks can also have more than two stages. Chung et al. present a three-stage solution, where the first stage generates phonetic networks, the second stage produces word network, and the third stage parses the word network using a natural language processor [14].

2.2 Graphone Models

In this section, we describe the training process for graphones, and their applications in speech recognition. We first discuss letter-to-sound (L2S) conversion because graphones were originally developed for the L2S task. L2S conversion is also an important part of vocabulary augmentation. We then discuss how graphones models can be incorporated directly into speech recognizers to construct hybrid speech recognition systems.

2.2.1 Graphone Letter-to-Sound Conversion

Graphone models capture the relationship between a word and its pronunciation by representing them as a sequence of graphone units. Each graphone unit has a grapheme sequence part and a phoneme sequence part. Maximum-likelihood learning on a pronunciation dictionary is used to automatically learn graphone units as well as transitions between graphones. In various publications, graphones have been referred to as joint-multigrams, graphonemes, and grapheme-to-phoneme correspondences.

The symbol sequences within A_k's can be concatenated to form a sequence R. We can do the same for the sequences within B_k's to form the sequence S. For the machine learning task, we observe the sequences R and S, and we want to estimate the underlying unit sequence G. In the L2S conversion context, R is the sequence of letters and S is the sequence of phonemes. The term graphone refers to g_k in this formulation. Table 2.1 shows some examples of graphone sequences for English words. The process of building a L2S converter is a three-step process. The first step is to develop a graphone model that best represents the data. The second step is to segment the training data into graphones. The third step is to build a N-gram

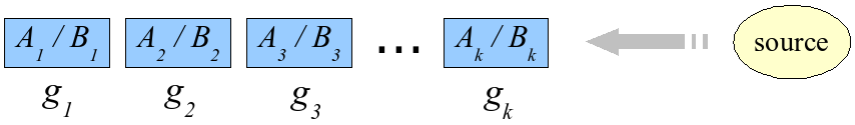

Figure 2-1: Generative probabilistic process for graphones as described by Deligne et al. According to an independent and identically-distributed (iid) distribution, a memoryless source emits units g_k that corresponds to A_k and B_k, each containing a symbol sequence of variable length.

Word	Graphone Sequence
alphanumeric	(alph/ae l f) (anum/ax n uw m) (eric/eh r ax kd)
colonel	(colo/k er) (n/n) (el/ax l)
dictionary	(dic/d ih kd) (tion/sh ax n) (ary/eh r iy)
semiconductor	(semi/s eh m iy) (con/k ax n) (duct/d ah kd t) (or/er)
xylophone	(xylo/z ay l ax) (phon/f ow n) (e/)

Table 2.1: Examples of graphone segmentations of English words. These maximum-likelihood segmentations are provided by a 5-gram model that allows a maximum of 4 letters/phonemes per graphone.

language model on the segmented training dictionary. This language model can then be used to perform L2S conversion on new words.

The theoretical foundation for graphone models is presented by Deligne et al., where they illustrate a statistical learning framework for aligning two sequences of symbols [16, 17]. The underlying assumption is that there is a memoryless source that emits the sequence of units $G = g_1, g_2, g_3, ..., g_K$, where g_k contains two parallel parts: A_k and B_k (see Figure 2-1). A_k and B_k are each variable-length symbol sequences themselves.

For the first step, the graphone model is learned through maximum-likelihood training. Expectation maximization (EM) is performed on the training dictionary by running the forward-backward algorithm on directed-acyclic graphs (DAG) representing letter and phoneme sequences. Each DAG corresponds to one word-pronunciation pair. It can be visualized as a 2-dimensional grid: one dimension represents the traversal of letters in the word, and the other dimension represents the traversal of

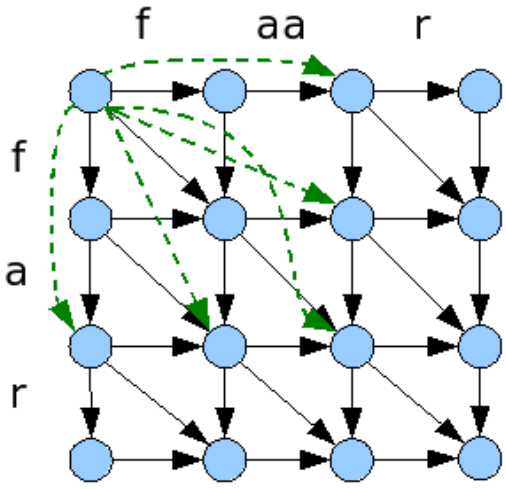

Figure 2-2: Directed acyclic graph representing possible graphone segmentations of the word-pronunciation pair (far, f aa r). Black solid arrows represents graphones of up to one letter/phoneme. Green dashed arrows (only a subset shown) represent graphones with two letters or two phonemes. Traversing the graph from upper-left corner to lower-right corner generates a possible graphone segmentation of this word.

phonemes in the pronunciation (see Figure 2-2). Each arc in the DAG corresponds to a graphone, and is labeled with the graphone's probability. During training, we must specify the minimum and maximum number of letters and phones allowed for each graphone, which limits the length of arcs in the DAG. Any path through this DAG represents a possible graphone segmentation of this word-pronunciation pair. The forward algorithm runs from the upper-left corner toward the lower right corner, while the backward algorithm runs the opposite way. These forward and backward scores are used to calculate the posterior for each arc. Posteriors corresponding to the same graphone are accumulated over DAGs of all dictionary entries, and then normalized to generate the updated probability for that graphone. This process is repeated until the likelihood of the training data stops increasing.

Mathematically, the EM training process can be formulated as follows. The likelihood of a dictionary, Λ, is the product of the likelihood of W word-pronunciation pairs in the training dictionary. For a word-pronunciation pair, let the letter sequence be $r_1, r_2, ..., r_M$, and the phoneme sequence be $s_1, s_2, ..., s_N$. We also denote a stream of graphones as $g_1, g_2, ..., g_K$. The likelihood of this word-pronunciation pair is the

sum of the likelihoods for all possible segmentations. The likelihood for a possible segmentation, z, is $p(z) = \prod_{k=1}^{K} p(g_k)$. The goal of the EM training goal is to find a graphone model that maximizes the likelihood of the training dictionary which is:

$$p(\Lambda) = \prod_{w=1}^{W} \Big(\sum_{z \in \{z\}_w} p(z) \Big) \tag{2.1}$$

where $\{z\}_w$ is the set of all segmentations of the w-th word in the dictionary.

The forward-backward algorithm is formulated as follows. Let $t_{m,n}$ represent the node at location (m, n) in the word-pronunciation pair's DAG, where $t_{0,0}$ is the upper left node in the DAG, and $t_{M,N}$ is the lower right node (see Figure 2-2). Let $\alpha(t_{m,n})$ be the forward score and $\beta(t_{m,n})$ be the backward score corresponding to node $t_{m,n}$. The forward score for $t_{m,n}$ represents the likelihood of letters and phonemes between $t_{0,0}$ and $t_{m,n}$. The backward score for $t_{m,n}$ represents the likelihood of the of the letters and phonemes between $t_{m,n}$ and $t_{M,N}$. For boundary conditions, we use $\alpha(t_{0,0}) = 1$ and $\beta(t_{M,N}) = 1$. Let $X_{m,n}$ denote an arc that leads into node $t_{m,n}$, and let $x_{m,n}$ denote the starting node of this arc. Let $Y_{m,n}$ denote an arc that leads out of node $t_{m,n}$, and let $y_{m,n}$ denote the ending node of this arc. In addition, $\{X_{m,n}\}$ denotes the set of all arcs that enter node $t_{m,n}$, and $\{Y_{m,n}\}$ is the set of all arcs that leave this node. As we move through the graph, the forward and backward scores can be computed using:

$$\alpha(t_{m,n}) = \sum_{\{X_{m,n}\}} \alpha(x_{m,n}) p(X_{m,n}) \tag{2.2}$$

$$\beta(t_{m,n}) = \sum_{\{Y_{m,n}\}} \beta(y_{m,n}) p(Y_{m,n}) \tag{2.3}$$

After the forward and backward scores are calculated, we can re-estimate the posterior probability for each graphone g_q used in the word-pronunciation pair w:

$$pos_w(g_q) = \frac{\sum_{(m,n) \in \{gr(X_{m,n}) = g_q\}} \alpha(x_{m,n}) p(X_{m,n}) \beta(t_{m,n})}{\sum_{m,n} \alpha(t_{m,n}) \beta(t_{m,n})} \tag{2.4}$$

where $gr(X_{m,n})$ is the graphone corresponding to the arc $X_{m,n}$.

The posterior probabilities for all graphones are accumulated through all word-pronunciation pairs in the training dictionary, and normalized such that graphone probabilities sum to one. In the following equation for updating the graphone probabilities, Q denotes the graphone set size.

$$p(g_q) = \frac{\sum_{w=1}^{W} pos_w(g_q)}{\sum_{w=1}^{W} \sum_{q=1}^{Q} pos_w(g_q)} \tag{2.5}$$

After updating probabilities for all graphones, we check if the training dictionary's likelihood has stopped increasing, and repeat another EM iteration if needed. The training dictionary's likelihood is calculated by taking the product of $\alpha(t_{M,N})$'s for all words. The number of graphones can be controlled in a number of ways, the simplest being trimming with a probability threshold. Also, in a practical implementation, the probabilities would be stored in the log domain to reduce the effect of machine precision.

For the second step of the graphone training process, the Viterbi algorithm is applied on the DAGs to segment the entire training dictionary into sequences of graphones. Running this algorithm on a word-pronunciation pair involves calculating the Viterbi scores for each node in its DAG and setting back points which can then be used to perform a backward trace to identify the graphones corresponding to the best segmentation. The Viterbi scores, $\gamma^*(t_{m,n})$, are updated while moving from the upper-left to the lower-right corner of the DAG. This score represents the probability of the best graphone segmentation up to node $t_{m,n}$. It can be calculated using the following equation:

$$\gamma^*(t_{m,n}) = \max_{\{X_{m,n}\}} p(X_{m,n}) \gamma^*(x_{m,n}) \tag{2.6}$$

The back-pointer for node $t_{m,n}$ is set to point to the previous node that corresponds to the maximum value in Equation 2.6.

For the third step, standard N-gram modeling is used to compute statistics over the segmented training dictionary, which includes smoothing and backing-off to shorter N-grams. These statistics can then be combined with standard search algorithms

(e.g. Viterbi search, beam search, Djikstra search) to decode an unknown word into its most-likely pronunciation.

In 2002, Bisani and Ney improve upon Deligne's formulation by adding an effective trimming method during training for controlling the number of graphones [6]. This is important because only a subset of the graphones that satisfy the letter/phoneme size constraints are actually valid chunks for modeling the training data. The intuition behind this is closely related to choosing morphemes that best represent the training data. Because only some of all possible chunks resemble valid morphemes or syllables, trimming can help us remove spurious graphones from our set. This study explores various constraints for the number of letter or phonemes allowed in a graphone, and finds that allowing 1-2 letters and and 1-2 phones per graphones, combined with a trigram language model gives the best performance in terms of phoneme error rate (PER). This trimming method, called evidence trimming, trims graphones based on how much they are used in segmenting words-pronunciation pairs in the dictionary, and is slightly different from trimming graphone probabilities. For a given iteration of EM, the evidence for a graphone refers to its accumulated posterior probability over all word-pronunciation DAGs. The graphone set is trimmed by applying a threshold for the minimum amount of evidence needed for a graphone to remain in the model. The probability of the trimmed graphones are then redistributed over the remaining graphones. Note that this is actually the opposite of the intuition from a language modeling point of view, where we use smoothing to assign probabilities to unseen data.

Both Bisani-Ney and Deligne et al. consider Viterbi training rather than EM training as a way to speed up the training process. During Viterbi training, only posterior probabilities for graphones along the best path in each DAG is accumulated. This could be a good approximation of considering all paths. However, both studies find that Viterbi training is highly sensitive to the initialization of graphone probabilities. One good initialization method is to initialize graphone probabilities proportional to each graphone's occurrence in all possible segmentations of the training dictionary.

In subsequent studies, Bisani and Ney investigate the use of graphones in large vo-

cabulary continuous speech recognition (LVCSR) [7]. Several studies by other authors also apply graphones to tasks such as dictionary verification [40]. Because graphone models are trained probabilistically, they are good for spotting errors in manually-annotated dictionaries. For entries that are likely to be erroneous, the probability associated with its graphone segmentation is very low compared to others. Using this information, we can automatically detect the mislabeled entries in the dictionary.

In a later study by Bisani and Ney, they revisit the L2S task with an revised training procedure that integrates EM training with language modeling (LM) [9], and produce results that are better than or comparable to other well-known L2S methods. EM training is first used to determine the set of graphone unigrams, in almost the same fashion as before. Then, the next-highest N-gram model (i.e. bigram) is initialized with the current model, and trained to convergence using EM. This process is repeated for successively-higher N-grams until the desired N-gram length is achieved. This differs from their previous study [6] in that EM is now performed for each N-gram length, rather than estimating the final language model directly on a dictionary segmented by unigram graphones. In the higher-order N-gram models, each node in the dictionary entry's DAG represents a position along the letter-sound sequence as well as the history for the N-gram that ends at this position. This means that unlike the older model described by Figure 2-1, the underlying probabilistic model is no longer a memoryless emitter. Instead, graphone emission probabilities are now conditioned on the previous $N-1$ emissions. Therefore, a node in the DAG is only connected to previous nodes that satisfy the N-gram history constraint. Each graphone N-gram is associated with an evidence value that is accumulated through all DAGs during each iteration of EM. Kneser-Ney smoothing is applied to graphone N-gram evidence values at the end of each EM iteration. Because these counts are fractional, we can't use discount parameter selection methods that assume integer counts. Powell's method is used to tune smoothing discount parameters by optimizing on a held-out set. When a graphone N-gram's fractional count falls below the discount parameter value, this N-gram is effectively trimmed. Another difference between this study and their previous study is that the new EM convergence criteria

is the likelihood of a held-out set rather than the training set, which could increase the generalizability of the L2S converter. This study concludes that training a long N-gram (i.e. 8 or 9-gram) along with tiny graphones (i.e. 0-1 letters/phonemes) produced the best L2S results. This finding is similar to the results produced by Chen with joint maximum entropy models [11], which is expected because the two methods have similar statistical learning processes. The results of this study are also comparable to the discriminative training results reported in [27].

2.2.2 Graphone Flat Hybrid Recognition

In speech recognition, graphones can be an effective method to mitigate the OOV problem in open vocabulary speech recognition. Ideally, OOV words could be recognized into a sequence of graphones that can be concatenated to form the correct spellings. There may be cases where an in-vocabulary word is recognized as a series of graphones, but as long as the graphone sequence still generates the correct spelling, the speech recognizer's output is still accurate. One way to move closer to this goal is to use a flat hybrid model for the speech recognizer's language model. To build this model, OOV words in the recognizer's training corpus are replaced by their graphone representations. Finding the graphone representation of a word can be done through the same process as L2S conversion, except we stop just before converting the most-likely graphone sequence to a phoneme sequence. From now on, we will refer to this process as *graphonization*. The only other part of the recognizer that needs to be modified is the lexicon, which needs to be augmented with a list of graphones and their pronunciations. This model is flat in the sense that graphones and words are treated as equals in the language model; the model is a hybrid because it contains statistics of N-gram transitions between graphones and words. Bisani and Ney have applied graphones in flat hybrid recognition for the Wall Street Journal dictation task [8]. Their graphonization process uses an N-gram size of 3, and their speech recognition's language model also has a N-gram size of 3. They find that at lower vocabulary coverage (88.8% and 97.4%), the word and letter error rates are significantly reduced by using this flat hybrid model. At high vocabulary coverage

(99.5%), this model does not degrade the recognition performance even though the added graphones could increase confusability with words. They also note tradeoffs regarding the graphone size. Larger graphones are easier to recognize correctly, but have worse L2S performance, leading to less accurate graphonization of OOV words in the training corpus. Small graphone sizes (i.e. 1-2 letters/phonemes) can also lead to more insertions in recognition output. In their experiments, they find that the trade-off point is a graphone size of 4 letters/phonemes for 88.8% and 97.4% coverage, and a graphone size of 3 for 99.5%.

Recently, Bisani-Ney's flat hybrid recognition have been combined with other methods. Vertanen combines graphone hybrid recognition with confusion networks and demonstrates improvement with the addition of the confusion network component [39]. This study also notes a technique for mitigating the tradeoff where bigger graphones are not as good in graphonization performance as smaller graphones. Instead of graphonizing the language model's OOV words directly, they first use a tiny graphone, large N-gram, L2S unit to generate pronunciations for OOV words, and then graphonize OOV words using bigger graphones with the L2S pronunciations as an additional constraint. Akbacak et al. use graphones to model OOV words in a spoken term detection task [1]. The recent applications of the graphone flat hybrid model show that this model is a start-of-the-art technique for speech recognition in domains where OOV words frequently occur.

2.3 SUMMIT Speech Recognition Framework

The SUMMIT Speech Recognizer is a landmark-based speech recognizer that uses a probabilistic framework to decode utterances into transcriptions [21]. The speech recognizer is implemented as a weighted finite-state transducer (FST), which consists of four stages: $C \circ P \circ L \circ G$. C and P encode phone label context and phoneme sequences. L is the mapping from phonemes to words, and G is the grammar or language model (LM). The acoustic model is based on diagonal Gaussian mixtures, and is represented by features derived from 14 Mel-Frequency Cepstral Coefficients

(MFCCs), averaged over 8 temporal regions. The mapping from phonemes to words are constructed directly from a pronunciation lexicon. The language model is an N-gram model trained on a large corpus of text, which is usually in the same domain as where the recognizer is used.

Decoding during recognition is performed through a two-pass search process. The first pass is a forward beam search where traversed nodes are labeled with their Viterbi scores. The second pass is a backward A* search where the Viterbi scores are used as the heuristic for the remaining distance to the destination. A wider search beam can produce more accurate results, but takes longer to run. To minimize computational complexity, the forward pass usually uses a lower-order N-gram than the backward pass. The backward A* search produces the recognizer's N-best hypotheses, along with their associated probability scores.

Chapter 3

Restaurant and Street Name Recognition

This chapter focuses on applying graphones to letter-to-sound (L2S) conversion of restaurant and street names. Since the end goal of L2S in a speech recognition system is to be able to correctly recognize the corresponding words, we evaluate the quality of the automatically-generated pronunciations through word recognition experiments. The objectives of these experiments are to find the optimal training parameters for the graphone model, determine how using multiple pronunciations affects performance, and verify that the graphone model generalizes well to truly unseen proper names.

3.1 Restaurant Data Set

The data set used in this experiment was originally collected for research reported in [13], in which subjects were asked to read 1,992 names in isolation. These names are selected from restaurant and street names in Massachusetts. They are representative of a data set that could be used for a voice-controlled restaurant guide application. These names are especially interesting because a significant portion of them are not in a typical pronunciation dictionary, so in order for the word recognizer to recognize these names, we must be able to automatically determine their pronunciations. In a typical system, as new place names are imported into the system through an update

process, a L2S converter can be run to automatically generate pronunciations for those words.

3.2 Methodology

The first step involves building graphone language models using the expectation maximization (EM) training process described in [9]. An established ∼150k-word lexicon used by SUMMIT is used for the training process. Different graphone models with maximum graphone size (L) of 1-4 letters/phonemes and graphone N-gram size (N) of 1-8 are explored. L is the upper bound on the maximum number of letters as well as the maximum number of phonemes in a graphone. There is no limit for the minimum number of letters or phonemes, except that we do not allow graphones to have neither letters nor phonemes (i.e. null). Note that it is important to allow 0-phoneme graphones because some letters can be silent. Also, for large-L models, the number of N-grams may no longer change after a certain N, and we do not continue building higher-order models beyond this point. These models are saved to files in ARPA language model format, which are then used to build L2S converters using the MIT Finite-State Transducer (FST) Toolkit [26].

The L2S converter is the composition of three FSTs: letter to graphones, graphone language model, and graphones to phonemes. The letter-to-graphone FST maps a letter sequence to all possible graphones that fit the letter sequence constraint. The graphone language model component is a direct FST translation of the ARPA language model file. The graphone-to-phoneme FST concatenates the phoneme components of the graphones. For a given word, a Viterbi forward search and A* backward search within the L2S FST is used to generate an N-best list for possible pronunciations. This L2S converter is then used to generate pronunciations for all 1,992 restaurant and street names.

For evaluation, a word-recognizer is built using the automatically generated pronunciations. A standard set of telephone quality acoustic models are used [43]. Utterance data collected from users are then used to determine the word-error rate (WER)

of this recognizer. These results are also compared to the same experiments using spellnemes.

3.3 Results and Discussion

3.3.1 Graphone Parameter Selection

Graphone parameters have a big impact on the performance of the system. Bigger graphones with larger L values can capture more information individually, but each word contains fewer graphones of this size. Smaller graphones on the other hand, capture less information individually, but because each word contains more graphones, we can more effectively use established language modeling techniques to make the model generalizable. Bigger graphones also lead to a much larger set of graphones compared to smaller graphones.

Graphone models with different L and N parameters were used to generate word pronunciations. The raw data for graphs in this section are located in Appendix A. Figure 3-1 shows the recognition word error rate (WER) for different combinations of L and N values. For lower N values, high-L models perform the best because low-L graphones capture little information individually. For high N values, low-L models perform better, most likely because the language modeling framework is more powerful and generalizable than the EM framework that identifies the graphone set. One explanation for this could be that unlike the Kneser-Ney discounting used with N-grams, there is no discouting that discounts larger graphones for smaller graphones. Numerical precision may also play a role. For large L, each graphone unigram has a very small probability, making it likely that some valuable graphones could be discarded when their evidence falls below numerical precision. For a sense of comparison, graphone set sizes for the unigram models are: 0.38k, 3.7k, 20k, 54k, for $L = 1, 2, 3, 4$ respectively. Numerical precision could also play a role for higher-order N-grams as well, but because the training process trains each N-gram model to convergence before ramping up to the next-higher N-gram, we have much more

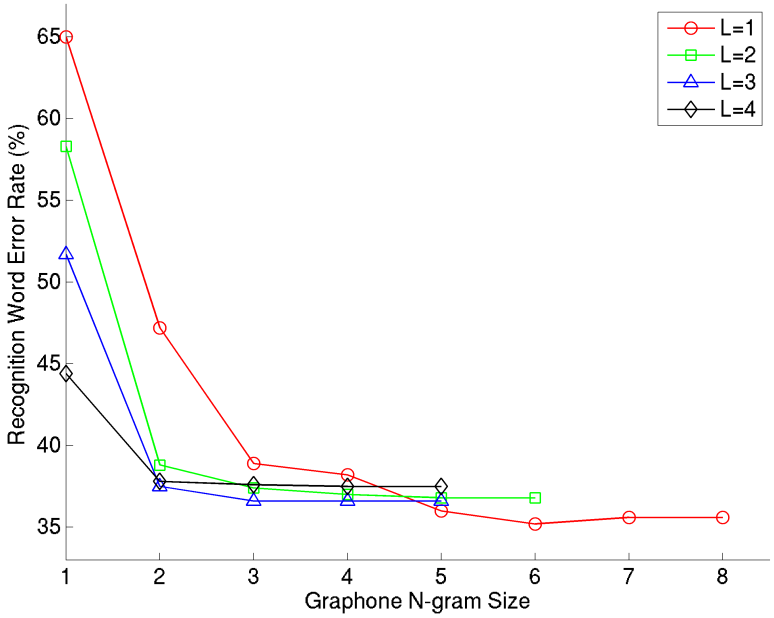

Figure 3-1: Word recognition performance using automatically generated pronunciations produced by a graphone L2S converter trained with various L and N values.

model stability as we increase N.

These results show that the best L-N combination is $L = 1$, and $N = 6$, which has a WER of 35.2%. For $L = 1$, as we increase N past 6, the WER actually starts to increase slightly. This could be due to over-training as we increase the model order too much. For $L = 2$, WER does not decrease beyond a 5-gram. This trade-off point is around a trigram for $L = 3$, and bigram for $L = 4$.

Comparing graphone results with spellneme results from [13] shows that the best graphone models outperform spellneme models. With spellnemes, the WER is 37.1% while the best graphone model yields a WER of 35.2%. This could be because for graphones, EM training with smoothing is run for both unigrams as well as higher-order N-grams. For spellnemes however, although N-gram statistics are computed on a subword-segmented dictionary, there is no additional learning process for re-adjusting the higher-order N-gram models.

Word	First Hypothesis	Second Hypothesis	Third Hypothesis
albertos	aa l b eh r t ow z	ae l b er t ow z	ae l b er t ow s
creperie	k r ey pd r iy	k r ax p r iy	k r ax p er iy
giacomos	jh ax k ow m ow z	jh ax k ow m ow s	jh ax k ow m ax s
isabellas	ih z ax b eh l ax z	ax s aa b eh l ax z	ih z ax b eh l ax s
renees	r ax n ey z	r ax iy z	r ax n iy z
sukiyummi	s uw k iy uw m iy	s uw k iy y uw m iy	s uw k iy ax m iy
szechuan	s eh ch w aa n	s eh ch uw aa n	s eh sh w aa n
tandoori	t ae n d ao r iy	t ae n d ao r ay	t ae n d ao r ax
valverde	v aa l v eh r df ey	v aa l v eh r df iy	v ae l v eh r df ey

Table 3.1: Examples of the L2S converter's top 3 hypotheses using the graphone language model.

3.3.2 Adding Alternative Pronunciations

The pronunciation dictionary can have multiple pronunciations for each word. In fact, a manually-annotated dictionary will definitely have multiple pronunciations for some words because words can be pronounced in different ways. For an automatically-generated dictionary, we could just use the top hypothesis for the L2S converter as the pronunciation. However, the next few hypotheses in the recognizer's output N-best list (ranked by decreasing probability) could be valuable as well, especially if a word truly does have multiple pronunciations. Interestingly, for restaurant and street names, multiple pronunciations can have another added benefit: people may pronounce names incorrectly because they are guessing based on prior language knowledge. For example, for a French restaurant name, some people may pronounce it closer to its French pronunciation, while others may say it by reading the name as if it were an English word. If the L2S converter suggests some of these commonly-guessed pronunciations as alternatives, then we are more likely to recognize place names correctly, even if the subject doesn't pronounce it correctly in the first place. This all works under the assumption that the L2S converter's top hypotheses are the ones that make the most linguistic sense based on the training dictionary. For the graphone-based L2S, this is usually the case when the training dictionary is large. Table 3.1 shows some examples of graphone L2S conversion using $L = 1$ and $N = 6$.

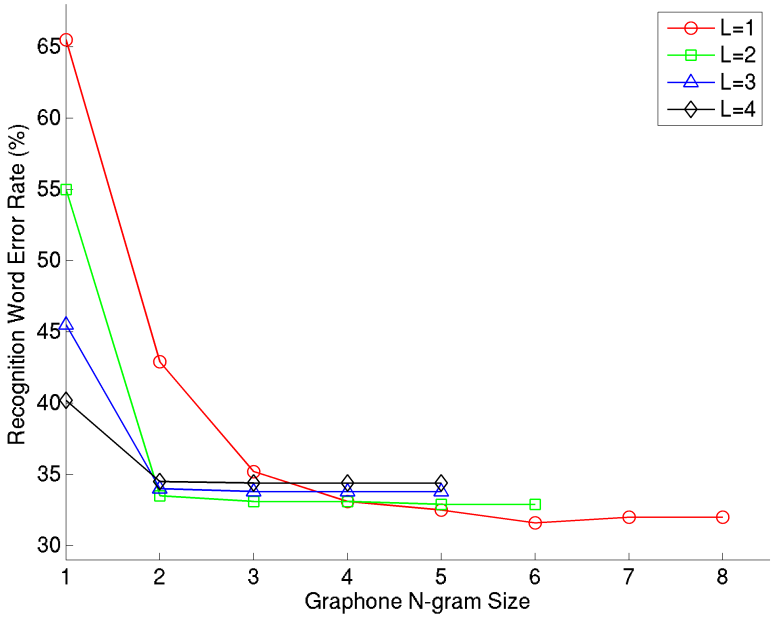

Figure 3-2: Word-recognition performance using top 2 pronunciations generated using various L and N values.

For most of the words, the first hypothesis is an excellent pronunciation. Sometimes, the quality of the pronunciation degrades rapidly, such as for *renees*, but for most words, the first few hypotheses are all reasonable approximations of the true pronunciation. Some of these variations account for variations in phonemes that are very close to each other, such as *s* and *z*, and *ax* and *ae*. These variations are also useful for how fast the person says the word. For *creperie*, the first two pronunciations are more accurate for someone saying it at conversational speed, especially if they know french, whereas the third pronunciation is in more enunciated form.

First, we explore various values of L and N for generating top 2 pronunciations for each word. The WER graph, shown in Figure 3-2, has a similar shape as the graph for using top pronunciation only. These results are slightly better than using the top 1 pronunciation. The best L-N combination is still $L = 1$ and $N = 6$, with a WER of 31.6%. Since for both top 1 and top 2 pronunciation cases, the same L-N combination gives the best results, we use this combination for experimenting with even longer lists of alternative pronunciations.

To study the effect of adding more alternative pronunciations, the number of pronunciations per word is varied between 1 to 50 for $L = 1$ and $N = 6$. Multiple pronunciations for each word are weighed equally in the recognizer's FST framework. We expect that as we increase the number of pronunciations per word, the word-recognition performance first gets better because we are adding more useful variation for each word, and then gets worse because too many pronunciations introduces too much confusion into the recognition framework, and dilutes the probabilities for the correct pronunciations. These results, along with a comparison to spellnemes are shown in Figure 3-3. We see the most recognition improvement when the number of pronunciations is increased from 1 to 3. We then have a trade-off point of about 10 pronunciations per word before the WER starts increasing with added pronunciations. These results indicate that using graphones are the most advantageous than spellnemes for using the top 1 hypothesis. Spellnemes do better at high number of included pronunciations, possibly because their lexically-derived nature allows the pronunciation quality to degrade more gradually as we move down the N-best hypothesis list. In addition, because spellnemes are more linguistically-constrained, they may be better at generating alternative pronunciations when applying phonological rules leads to multiple correct pronunciations that are distant from each other.

3.3.3 Generalizability of Graphone Models

Finally, we explore the generalizability of graphone L2S for this task. The restaurant and street name data set includes some common English names that are likely to occur in the training dictionary. The set of truly unseen names provide a good test for the generalizability of the model. Out of the 1,992 names, 626 names have an exact match in the dictionary used to train the graphone model, and the remainding 1,366 are considered to be unseen. In order to generate good pronunciations for unseen words, the L2S converter must use the probabilities learned from the dictionary to guess the pronunciation of the unseen words. Also, this task is more challenging than evaluating L2S performance by dividing a dictionary randomly into train and test sets, because the unseen words for this task are names of places. Some examples

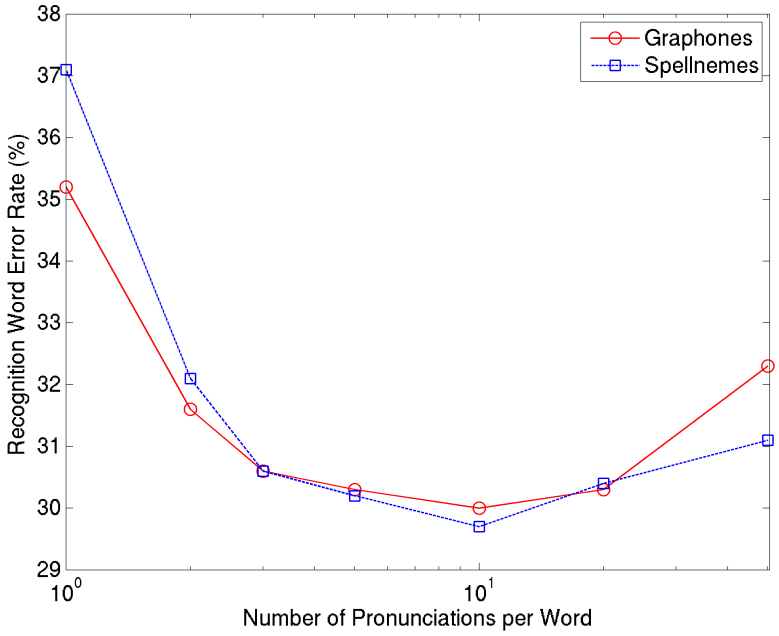

Figure 3-3: Word recognition performance with various number of pronunciations generated using the graphone ($L = 1$, $N = 6$) and spellneme L2S converters.

of words in the unseen set are: *galaxia*, *isabelles*, *knowfat*, *verdones*, and *republique*. Many of these words are foreign words, concatenation of multiple words, or purposely-misspelled words.

For using the top pronunciation only, WER results separated by seen and unseen words are shown in Figures 3-4 and 3-5. For unseen words, we see that we are still getting very good performance that are sometimes even slightly better than seen words. One reason for this favorable performance is that the EM training process has integrated smoothing. For unseen words, we also see a large dip at $L = 1$ and $N = 6$, which indicates possible overtraining after the N-gram length goes past 6. This dip is not present for the seen words.

For using the top 2 pronunciations, WER results for seen and unseen words are shown in 3-6 and 3-7. It's interesting to see that both plots have dips in their WER curves that indicate overtraining for very high N-grams. One explanation for this is that many seen words in the training dictionary only have one pronunciation. While the top L2S pronunciation may match the actual seen pronunciation exactly, the

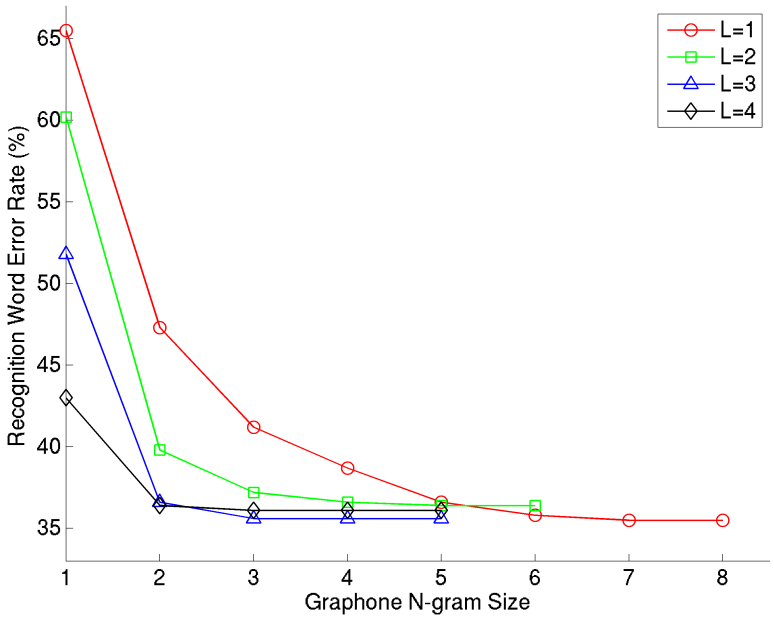

Figure 3-4: Word recognition performance for seen words when the top 1 pronunciation is used.

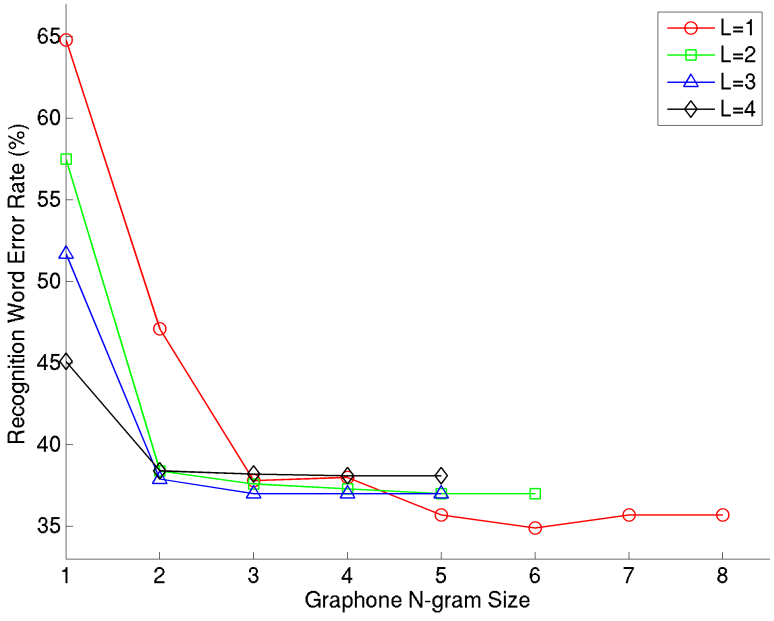

Figure 3-5: Word recognition performance for unseen words when the top 1 pronunciation is used.

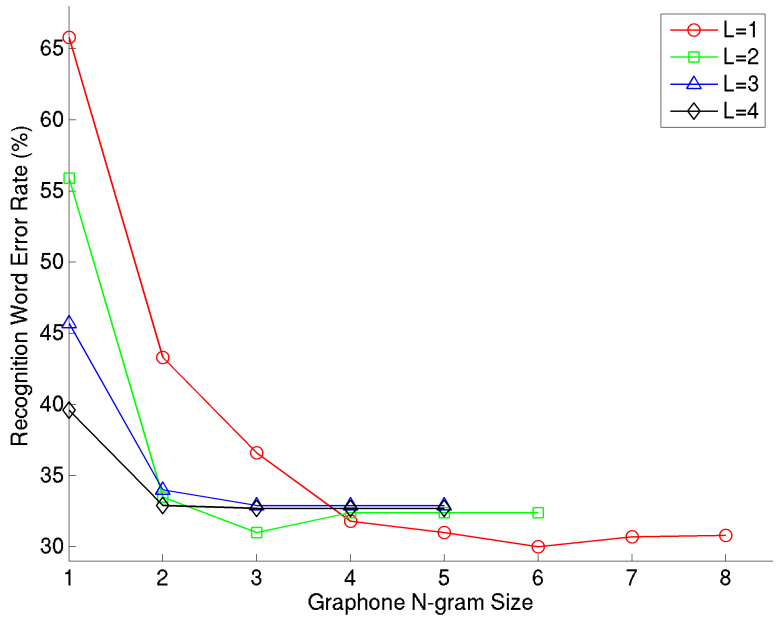

Figure 3-6: Word recognition performance for seen words when the top 2 pronunciations are used.

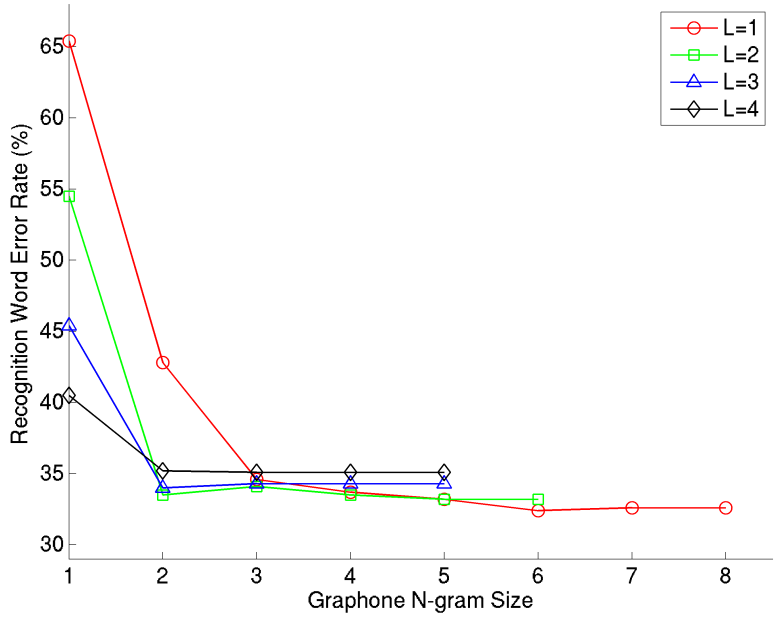

Figure 3-7: Word recognition performance for unseen words when the top 2 pronunciations are used.

second L2S pronunciation is only a hypothesized guess since it is not available in the training data. Finally, we have also shown good generalizability to unseen words for top 2 pronunciations, since the curves are similar to the ones for seen words.

3.4 Summary

In this chapter, we built a system that recognizes restaurant and street names, and demonstrated that graphone models are excellent for L2S conversion of these words. We further increased the word recognition performance of the system by incorporating more than one pronunciation for each word, using N-best hypotheses from the L2S converter. Finally, we showed that graphones generalize well to unseen words because their performance does not deteriorate for words not in the graphone training dictionary.

Chapter 4

Lyrics Hybrid Recognition

This chapter discusses applying graphones to hybrid recognition in the lyrics domain. Song lyrics often have many terms that are not actually English words. This provides a good environment to evaluate the performance of hybrid recognition using graphone language models. Graphone-based hybrid recognizers are built for various vocabulary coverages levels, and evaluated on spoken lyrics data. The results of this chapter are applicable for building a recognizer that could be integrated into a mobile or in-car entertainment system that allows users to query for songs by speaking their lyrics. The system constructed in this chapter also provides a foundation for moving to a truly open vocabulary task in the lecture domain, as described in the next chapter.

4.1 Lyrics Data Set

The lyrics data was collected in a previous experiment that focused on hybrid recognition using spellnemes, as well as query songs by lyrics [12]. The data collection process involved 20 subjects (13 males and 7 females), who were presented with 30-second clips of songs. They were instructed to speak any segment of the lyrics for that song based on what they heard. The subjects were also asked to type what they said, which serves as the reference for speech recognition experiments. A total of 1k songs were used in the data collection process. These songs were chosen from a set of ∼37k songs while ensuring that it was not too difficult to recognize the lyrics from

listening to these songs. Lyrics for these songs were scraped from *www.lyricwiki.org*, and were cleaned by removing non-ASCII characters as well as foreign songs. A foreign song identification script was written to recognize foreign songs by calculating the proportion of non-English words in the song, and using a simple threshold as a the classification criteria. The few remaining foreign songs were manually removed from the data set. Words in the corpus were also cleaned by correcting misspellings of words and splitting hyphenated words. These utterances can be harder to transcribe than normal dictation speech because lyrics can contain popular culture slangs that are not used in dictation speech. Using the same data for graphone experiments also allows us to make some comparisons between spellnemes and graphones for hybrid recognition.

4.2 Methodology

The lyrics of ∼37k songs are used as the language model training corpus for the recognizer. First, artificial vocabulary coverage levels are generated by ordering all words in this corpus by decreasing frequency, and then choosing the set of most occurring words that provide the desired vocabulary coverage level (i.e. such that x% of all corpus words, including duplicate words, is in the vocabulary). Depending on the vocabulary coverage, it is possible that some in-vocabulary words actually do not occur in the SLS pronunciation dictionary (∼150k words), so their pronunciations are initially unknown. In these cases, one pronunciation per word is generated using a letter-to-sound (L2S) converter with $L = 1$ and $N = 8$, which are generally the best parameters for L2S conversion [9]. This converter is trained on the SLS dictionary. Out of 46k unique words in the corpus, pronunciations of ∼15k words are automatically generated, and the rest are copied directly from the SLS dictionary.

The next step involves segmenting out-of-vocabulary (OOV) words from the corpus into graphones. This is done by training a letter-to-graphone (L2G) converter, with graphone size $L = 4$, using a 5-gram, on the SLS dictionary. This converter is almost the same as the letter-to-sound (L2S) converter, except we do not include the

Vocabulary Size	Training Coverage	Test Coverage
68	50.0%	53.0%
120	60.0%	63.5%
233	70.0%	74.3%
492	80.0%	84.5%
1443	90.0%	93.4%
1942	92.0%	94.9%
2766	94.0%	96.4%
4386	96.0%	97.7%
8572	98.0%	99.0%
14532	99.0%	99.4%

Table 4.1: Vocabulary size of hybrid recognizers, and their vocabulary coverage levels on the training and test sets.

FST that converts graphones into phonemes. OOV words in the training corpus are replaced by their L2G output. The language model for the recognizer is then trained on the resulting flat hybrid corpus. The set of $L = 4$ graphones are also added to the recognizer's vocabulary. The rationale for choosing $L = 4$ is that this has yielded the best results in Bisani and Ney's hybrid Wall Street Journal recognition task [8]. We have also conducted preliminary experiments with $L = 3$ or 4, which showed that $L = 4$ produces better results. The acoustic models for these recognizers are trained on telephone speech [43].

Several hybrid recognizers are built with training set vocabulary coverage levels ranging from 50% to 99% (see Table 4.1). These recognizers are evaluated on the 1k utterances from the lyrics data set through two methods: replacing each graphone sequence in the recognizer's output hypotheses by an OOV tag, and replacing graphone sequences by the concatenation of the graphones' letter sequences. Replacing by an OOV tag gives us an understanding of how graphones can prevent misrecognition of words adjacent to an OOV term. Concatenating graphones gives us a sense of how well the graphones spell out OOV words. Word-only recognizers are also trained on the same vocabulary coverage levels for comparison with hybrid recognizers. Finally, a word-only recognizer with 100% training corpus coverage is built as well.

Since this recognizer is geared toward song retrieval, most of the test utterances are also in the recognizer's language model training corpus. The only situation where

the test utterance's reference transcription differs from the training corpus is when the user incorrectly hears the song's lyrics, which did happen during data collection. This factor causes the training corpus to have a 0.2% OOV rate on the test set. Nevertheless, because most of the phrases in test utterances are in the training data, this task is mostly a closed-vocabulary task, and should perform better than completely open-vocabulary tasks.

We evaluate the recognition performance on word error rate (WER), sentence error rate (SER), and letter error rate (LER). WER tells us how many words we have exactly correct in our recognition output, but can be overly harsh for words that the graphones spell almost correctly. Therefore, we also evaluate our results on LER to reward the recognizer for spelling words close to correct. SER tells us if we get entire utterances correct, and is the most strict measure of correctness used for this study.

4.3 Results and Discussion

The results of this experiment show dramatic improvements for combating OOV word by using a graphone flat hybrid model. The raw data for all graphs in this section is located in Appendix B. Figure 4-1 shows WER recognition results for three setups. The first setup is the baseline word-only recognizer that only knows words within the specified coverage level. We see improvement in WER when we use a hybrid recognizer and replace graphone sequences in the output by OOV tags. This improvement shows that graphones are useful as an OOV detector, and helps us prevent words around OOVs from being misrecognized. Next, we see even more improvement when we concatenate graphones to spell OOV words. This improvement indicates that in this task, graphones are effective in hypothesizing pronunciations of OOV words. We get the most improvement for low vocabulary levels (50%), where using graphones can improve WER by almost 40 percentage points. For very high vocabulary (99%), there are still some performance gains for using the hybrid recognizer. The word-only recognizer with 100% training vocabulary coverage yields a WER of 28.4% (see Table B.1), which is about equivalent to the performance of a hybrid recognizer with

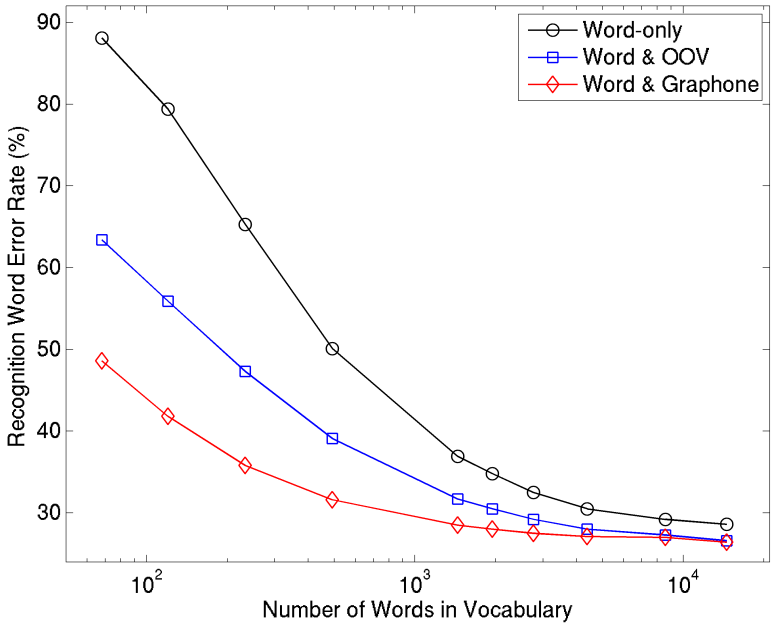

Figure 4-1: Recognition word error rate (WER) for recognizers with 50-99% training vocabulary coverage. Word & OOV is replacing each output graphone sequence by an OOV tag. Word & Graphone is concatenating consecutive graphones in the output to spell OOV words.

90% coverage. Table 4.2 shows some examples of hybrid recognition output for 92% vocabulary coverage level.

The sentence error rates (SER) for the three approaches are shown in Figure 4-2. Here, replacing graphones by the OOV tag performs about the same as word-only recognition, except at very high vocabulary coverage. This means that at low vocabulary coverages, there are just too many words that are OOV to see any improvement for using graphones. However, at very high vocabulary coverage, some in-vocabulary words that would have been misrecognized by the word-only recognizer, possibly because they have low probability, are actually recognized correctly as entire words when the graphone model is added. We also see dramatic improvement when we concatenate graphones to spell OOV words, which allows us to correctly recognize some sentences with OOV words. Also, the 100% coverage word-only recognizer has a SER of 67.5% (see Table B.1).

Finally, letter error rates (LER) are computed for these experiments (see Figure

Word-Only Recognizer	Hybrid Recognizer	Reference
the tide at the gate	the time to [hesi|tate]	the time to hesitate
tonight this could you send until the night	[tenn|ant|cisc|o] days san [fran|cisc|o] nights	san francisco days san francisco nights
in the l bum of my memory	in the [alb|um] of my memory	in the album of my memory
i can see you water and moonlight be missing	i can see you water and moonlight [beam|ing]	i can see water and moonlight beaming
she ring cloud it in crowd	[ch|eer|ing] clouds in [glit|ter|ing|crow|ds]	shimmering clouds glittering crowds
a long story short and your situation	a [con|stan|t|sour|ce] of your [fru|stra|tion]	a constant source of your frustration

Table 4.2: Example recognizer output at the 92% vocabulary coverage level. Square brackets denote concatenated graphones (only letter part shown). Vertical bars are graphone boundaries.

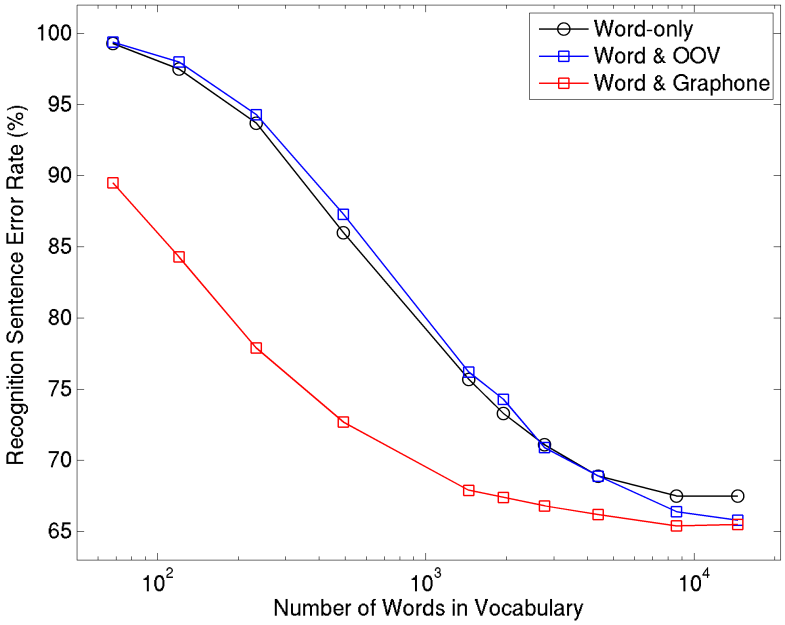

Figure 4-2: Recognition sentence error rate (SER) for recognizers with 50-99% training vocabulary coverage. Word & OOV is replacing each output graphone sequence by an OOV tag. Word & Graphone is concatenating consecutive graphones in the output to spell OOV words.

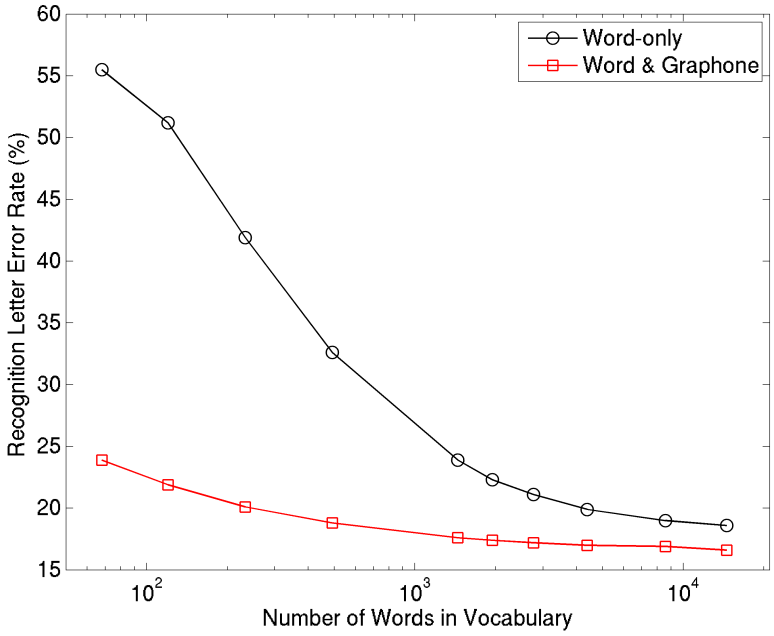

Figure 4-3: Recognition letter error rate (LER) for recognizers with 50-99% training vocabulary coverage. Word & Graphone is concatenating consecutive graphones in the output to spell OOV words.

4-3). On this plot, we do not show results for replacing by the OOV tag because this approach actually doesn't make sense with LER; we do not know how many letters the OOV tag truly encapsulates. For LER, we also see dramatic improvements in how much graphones can help spell out words correctly. Because the curve for using graphones is much flatter than the graph for WER, we can conclude that many words at low vocabulary coverages are actually spelled very close to its true spelling. The 100% word-only recognizer has a LER of 18.5% (see Table B.1), which is similar to the LER performance of the 80% hybrid recognizer with only a 492-word vocabulary.

We can also compare these graphone results with the spellneme hybrid results reported in [12]. In the spellnemes study, a hybrid recognizer is built by converting OOV words to spellnemes, just as in graphone hybrid recognition. Although WER and SER results are reported for word-only and replacing spellneme sequences by an OOV tag, no results are reported for concatenating spellnemes together. For the replacing subwords by OOV approach, graphones perform better by about 10 percentage points for low vocabulary coverage. This advantage gradually decreases

as we increase the vocabulary coverage, and at very high vocabulary coverage, the two methods perform about the same in WER. For SER of replacing by the OOV tag, spellnemes perform about 10 percentage points better than graphones at low vocabulary coverages, and perform about the same as graphones at high vocabulary coverage. One reason could be that at very low vocabulary coverages, graphones are more eager to appear at places where an in-vocabulary word should appear, but may still end up with the same spelling anyways. However, these sentences are still penalized because of they contain OOV tags. Therefore, WER is a better measure of performance than SER for this task.

4.4 Controlling the Number of Graphones

Controlling the size of the graphone set can be important for applications where the size of the recognizer's lexicon is limited. Because graphones are added directly to the recognizer's lexicon, they can also increase the size of the resulting recognizer. With the untrimmed $L = 4$ graphone model, 22k graphones are needed to segment all words in the training corpus. The L2G model used for this segmentation has 54k unique graphones. Thus for the 50% coverage vocabulary, although only 68 words are in-vocabulary, almost 22k graphones are needed to model the OOV words. For high vocabulary coverages, the number of graphones are considerably less because less words are considered OOV in the training corpus. Nevertheless, it is beneficial to investigate trimming the graphone set.

The size of the graphone set is reduced by modifying the procedure for training graphones on the pronunciation dictionary. First, the unigram EM is run to convergence with the Kneser-Ney evidence discount parameter automatically chosen by Powell's method. For $L = 4$, this discount value is 0.07. We then run the unigram EM again with a fixed discount value to trim the graphone set to the desired value. For a sense of comparison, graphone evidence values accumulated over the entire training dictionary can be as big as 5k to 10k for common graphones like (s/s), (er/er), and (ing/ih ng); and as small as allowable by machine precision. Finally, we use the

Discount Value	Number of Graphones
0.07	54k
3.0	11k
5.0	7k
8.0	5k
10.0	4k
20.0	2k

Table 4.3: Number of graphones for $L = 4$ unigram model after trimming using the specified Kneser-Ney discount value.

	Full Graphone	4k Graphone	2k Graphone	Word-Only
SER	72.7%	75%	78.0%	86.0%
WER	31.6%	34.0%	37.1%	50.1%
LER	18.8%	20.3%	22.5%	32.6%

Table 4.4: Comparison of recognition performance for the full graphone hybrid model and trimmed graphone models (4k and 2k graphone set size). We also show results for the word-only recognizer. All of these recognizers have 80% vocabulary coverage.

trimmed model to ramp up to higher graphone N-grams using same procedure as before. Table 4.3 shows how the size of the graphone set changes as the evidence discount parameter is varied.

For the lyrics experiments, we ramp up the 4k and 2k trimmed unigram graphone models to 5-grams, and then use these models to segment OOV words at 80% vocabulary coverage. Table 4.4 shows results for concatenating graphones to spell OOV words. These results indicate that trimming the graphone set does degrade the hybrid's recognition performance, but this effect is still small compared to the performance of the word-only recognizer. Even after reducing the number of graphones used in the hybrid model by 10-fold to 2k, we still achieve recognition performance that is significantly better than word-only recognition.

4.5 Summary

In this chapter, we applied graphone hybrid recognition to music lyrics utterances. We demonstrated significant performance improvements over the word-only recognizer at

all vocabulary coverage levels tested. By replacing graphone sequences with OOV tags, we showed that using the hybrid recognizer can reduce errors for in-vocabulary neighbors of OOV words. By concatenating graphone sequences to spell OOV words, we improved recognition performance even further, demonstrating that graphones can correctly spell OOV words. Finally, we explored trimming the graphone set during EM training, and showed that even with a much smaller graphone set, the hybrid recognizer still significantly outperforms the word-only recognizer.

Chapter 5

Spoken Lecture Transcription

This chapter explores using graphone hybrid recognition for spoken lecture transcription. This task is an open-vocabulary task because we cannot anticipate everything the lecturer says in the recorded audio, especially if it's an unknown topic. Our main objective is to build a hybrid recognizer and compare this recognizer's performance with a word-only recognizer at various vocabulary coverage levels. We also explore several extensions for improving the hybrid recognizer, including trimming the graphone set, using a full vocabulary with the hybrid model, and hierarchical models.

5.1 Lecture Data Set

The lecture data is mainly a set of manually-transcribed lectures from the MIT OpenCourseWare and MIT World databases. The data was originally used in [22] to develop a lecture transcription engine so that the lecture text can be easily accessed from a web interface. The training corpus for the speech recognizer has 480k utterances comprised of 6.8M words. It not only contains transcribed lectures, but also contains data from the Switchboard corpus and the Michigan Corpus of Academic Spoken English (MICASE). There are a total of 49.3k unique words in this training corpus, which would be the vocabulary size of a word-only recognizer with 100% training corpus coverage. The test set has 7.4k utterances containing 72k words, selected from lectures on computer algorithms, speech recognition systems, biology, and differential

equations. The entire training corpus has a 99.4% vocabulary coverage on the test corpus.

The lecture data provides us with a practical application that has good opportunities for working with OOV words that are likely to be important. In these lectures, the OOV words are often technical terms that are crucial for the understanding of the lecture. Users also are likely to use these terms as keywords for searching through the lectures transcript. Therefore, the ability to hypothesize spellings for these OOV words can significantly increase the value of the lecture-browsing application.

5.2 Methodology

A baseline word-only recognizer and a hybrid recognizer are built for several vocabulary levels, ranging from 88% to 99% coverage on the *training* corpus. Most of the steps for building the recognizers are analogous to those for the lyrics recognizer described in the Chapter 4. In-vocabulary words for each coverage level are selected by decreasing frequency in the lecture training corpus. Table 5.1 shows the vocabulary sizes of the hybrid recognizers, as well as the corresponding vocabulary coverage levels on training and test sets. Finally, we also build a word-only recognizer using 100% of the training vocabulary.

For building a word-only recognizer, pronunciations are needed for all words that are considered to be in-vocabulary. Most of the 49.3k words in the training dictionary are found in the SLS pronunciation dictionary, but 4.2k of those words are missing pronunciations. For those words, a graphone letter-to-sound converter ($L = 1$, $N = 8$, trained on the SLS dictionary) is used to automatically generate one pronunciation for each word. For the hybrid recognizer, OOV words in the training corpus are replaced by their graphone representations using a letter-to-graphone (L2G) converter, which is trained on the SLS dictionary with $L = 4$ and $N = 5$. These graphones are also added to the recognizer's lexicon. For both word-only and hybrid models, we use the SRI Language Modeling Tookit (SRILM) [36] to build their recognizer language models. For this step, a trigram model is built on the training corpus using Witten-

Vocabulary Size	Training Coverage	Test Coverage
1320	88.0%	85.6%
1780	90.0%	88.4%
2468	92.0%	90.4%
3594	94.0%	93.0%
5661	96.0%	95.0%
10633	98.0%	97.2%
17374	99.0%	98.4%

Table 5.1: Vocabulary sizes for the hybrid recognizer, and corresponding vocabulary coverage levels on the training and test sets.

Bell smoothing and a N-gram pruning threshold of 0.000001. The acoustic models for this experiment are the same as those in the original lecture transcription recognizer, which is trained from about 121 hours of speech from MIT lectures. This recognizer also already has models for common non-speech utterances such as <um>, <laugh>, <cough>, etc.

These recognizers are tested on new data from lectures not in the training set. Recognition performance are measured by computing word error rate (WER), sentence error rate (SER), and letter error rate (LER) for each recognizers's output. For LER, all spaces in the recognizer's output and the reference text are replaced by a special token to take into account of whether consecutive words are separated correctly in the recognizer's output.

5.3 Results and Discussion

The hybrid model performs well when compared to word-only recognizers with the same vocabulary coverage. The raw data for all graphs in this section is available in Appendix C. Figure 5-1 shows WER for a) the word-only recognizer, b) replacing each graphone sequence in the recognizer output with an OOV tag (Word & OOV), and c) concatenating consecutive graphones in the output to spell out OOV words (Word & Graphone). The trend here is similar to the ones depicted in Chapter 4 for lyrics recognition. The difference between word-only and replacing by an OOV tag shows that graphones help us reduce mistakes near OOV words. The difference

Word-Only Recognizer	Hybrid Recognizer	Reference
whatever they're using job that now i guess <uh> to do way lower	whatever they're using [java] now i guess <uh> to do [eul|er]	whatever they're using java now i guess to do euler
these different kinds of like rocks old and hydrogen	these different kinds of a [hydr|ox|al|s] and [hydr|ogen|s]	these different kinds of hydroxyls and hydrogens
artistic models are made out of <uh> gauss in mixture switch	are [acou|stic] models are made up of <uh> [gaus|s|ian] makes [ture|s] which	our acoustic models are made up of <uh> gaussian mixtures which
the right missiles are responsible for proteins and this is	the [ribo|some|s] are responsible for protein [syn|thes|is]	the ribosomes are responsible for protein synthesis
individual molecules and the scientist kelvin	individual molecules and the [cyto|skel|eton]	individual molecules and the cytoskeleton

Table 5.2: Example recognizer output for the hybrid and word-only recognizers at 96% training corpus coverage, along with reference transcriptions. Square brackets indicate concatenation of graphones (only letter part shown). Vertical bars denote graphone boundaries.

between replacing by an OOV tag and concatenating graphones shows that we can spell some OOV words correctly using graphones. The performance benefit of using graphones diminishes as we increase the recognizer's coverage. The more coverage we use, the less OOV words we have to segment into graphones, causing the hybrid language model to have less data about graphone-to-graphone transitions, as well as word-to-graphone transitions. Table 5.2 shows some examples of the hybrid recognizer correctly or almost-correctly spelling OOV words. This hybrid recognizer has 96% training corpus coverage, equivalent to 95% test corpus coverage. Also note that sometimes the hybrid recognizer will recognize an in-vocabulary word as a sequence of graphones, but usually with the correct spelling.

The comparison for SER is shown in Figure 5-2. One trend is that replacing by the OOV tag consistently does slightly worse than word-only recognition. This could mean that some in-vocabulary words are consistently recognized into their equivalent graphone form. Replacing by the OOV tag does not turn these words back to their original spelling, so the sentences are counted as incorrect. However, after

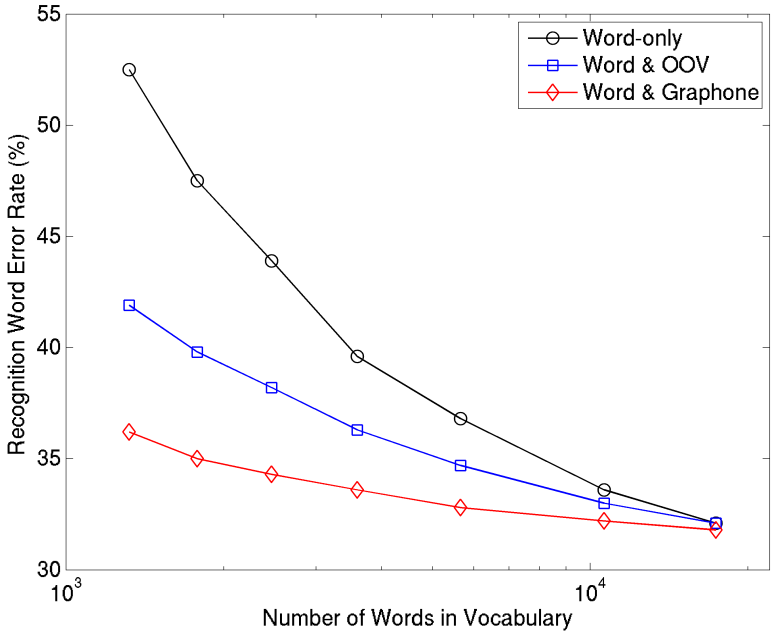

Figure 5-1: Recognition word error rate (WER) for 88-99% training vocabulary coverages. Word & OOV is replacing each output graphone sequence by an OOV tag. Word & Graphone is concatenating consecutive output graphones to spell OOV words.

the graphones are concatenated into words, almost all of these words are correctly spelled back into their original form.

By inspecting the hybrid recognizer's output, we notice that the hybrid recognizer is very good at spelling words that are also in the training corpus, but are considered as OOV based on the chosen vocabulary coverage. Since these graphone N-grams exist in the training corpus, the recognition process is much easier. For OOV words that do not occur in the training corpus, the task of spelling them with graphones is much harder. In most cases, the recognizer only produces a partially-correct spelling for these OOVs, which is counted as incorrect when calculating WER. To reward partially-correct spellings, LER is calculated for word-only and hybrid recognizers, as shown in Figure 5-3. The LER results show significant improvements for using the hybrid model. The curve for the hybrid model is also very flat, which indicates that even at lower vocabulary coverage levels, graphones are still effective in spelling many words mostly correctly.

Finally, we also note that none of these hybrids outperform a word-only recognizer

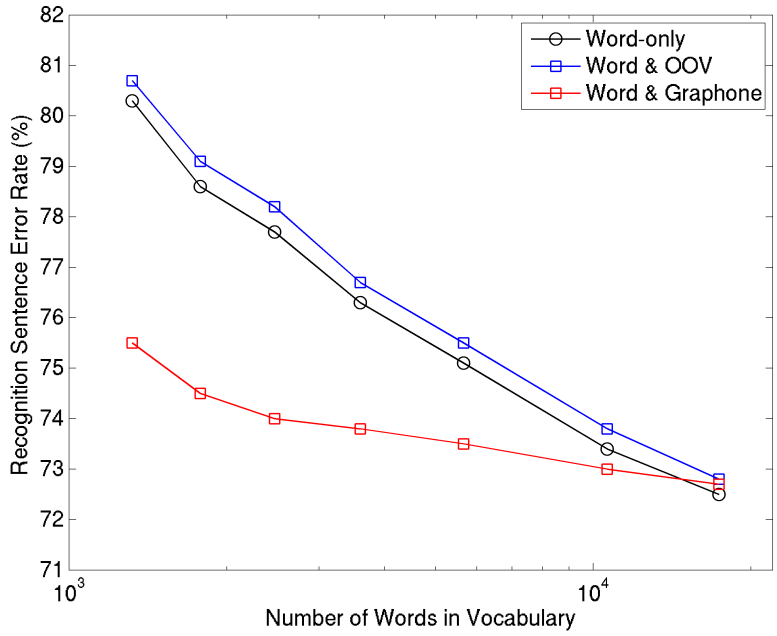

Figure 5-2: Recognition sentence error rate (SER) for 88-99% training vocabulary coverages. Word & OOV is replacing each output graphone sequence by an OOV tag. Word & Graphone is concatenating consecutive output graphones to spell OOV words.

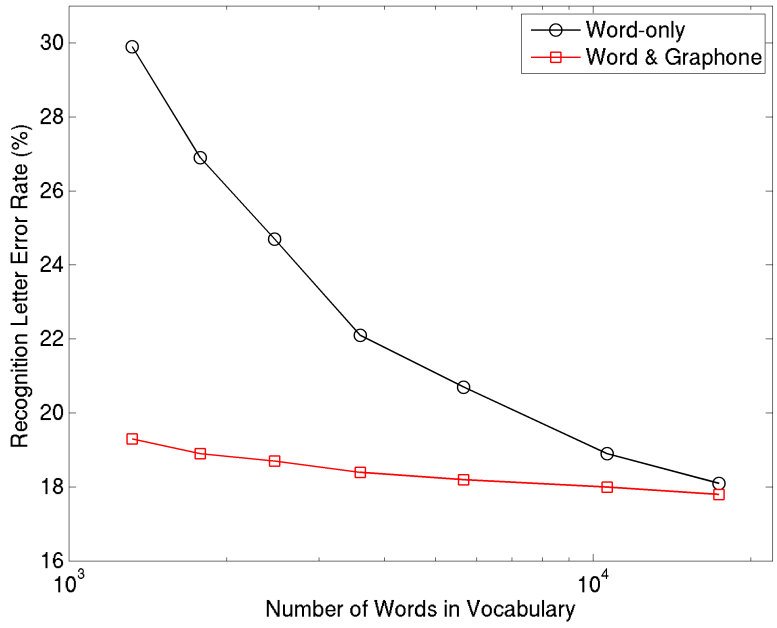

Figure 5-3: Recognition letter error rate (LER) for 88-99% training vocabulary coverages. Word & Graphone is concatenating output graphone sequences to spell OOV words.

with 100% training corpus vocabulary coverage, which includes 49.3k words with 4.2k pronunciations automatically generated by graphone L2S. As shown in Table C.1, the performance measures for this full-coverage word recognizer are: WER of 31.1%, SER 72.0%, and LER of 17.6%. The 99% hybrid has a WER 31.8%, SER 72.7%, and LER 17.8%. The 98% hybrid has a WER of 32.2%, SER 73.0%, and LER 18.0%. One factor could be that although the training and test corpora are from different lectures, the training corpus can still cover most of the words in the test corpus, so truly OOV words are not abundant in the test corpus. In fact, only 204 of the 4k unique words in the test set are not in the training corpus. Occurrences of these words account for 440 of the 72k total words in the test set. In this case, running with a large-vocabulary recognizer can offer more constructive constraints in the search space. The results however, are still very close between the full-coverage word-only recognizer and high-coverage hybrid recognizers.

Although the measured performance of hybrid recognizers is not better than the full-coverage word-only recognizer, there are still important advantages for choosing the hybrid approach. One advantage is vocabulary size. For the 98% hybrid recognizer, even with the graphones added, the recognizer's lexicon is still 20k smaller than the full-coverage word recognizer's, which can offer computational resource advantages. The big difference in vocabulary size is not limited to the lectures domain because data sparsity is an issue for many domains. The hybrid model also has the advantage of hypothesizing spellings of OOV words, rather than recognizing it into a completely different in-vocabulary word. This is important if the OOV word is a keyword for the lecture. In Table 5.3, we show some examples of 100% word-only recognition output and the 98% coverage hybrid, especially when the hybrid model's ability to hypothesize arbitrary words is useful. In addition, the hybrid model maybe be more useful if the training corpus is more limited. In this task, we have the luxury of training with a fairly large training corpus of transcribed lectures, but such corpora are not available for many applications. With a smaller corpus, even if we get pronunciations for all the words in the corpus, we still may not be able to account for a significant number of words in the test data, in which case open-vocabulary

Word-Only Recognizer (100% Coverage)	Hybrid Recognizer (98% Coverage)	Reference
linda was on great got their names attach to a version of this algorithm	[lind|ouzo] and great got their names attach to a version of this algorithm	linde buzo and gray got their names attached to a version of this <uh> algorithm
the outside it molecule	the [leph|atic] molecule	the allophatic molecule
this kind of reaction justification is the kind of linking	this kind of reaction [ysti|fica|tion] is the kind of linking	this kind of reaction esterification is the kind of linkage
in the cepstral domain we see that disputes	in the [ceps|tral] domain we see that the speed [s]	in the cepstral domain we see that the speech
the mitochondrial d n a from snort meander full bones	the mitochondria [l] d n a from [son] art [nea|nder|thal] bones	the mitochondrial d n a from <partial> <partial> neanderthal bones
here's the tube lists are also once again here's the phosphate	here's the two [glyc|er|oles] once again here's the phosphate	here's the two glycerols once again here's the phosphate

Table 5.3: Examples of recognition output for 100% coverage word-only recognizer and a 98% coverage hybrid recognizer, along with the corresponding reference transcriptions. Square brackets denote concatenated graphones. Vertical bars show graphone boundaries. Note that this table differs from Table 5.2 in that the hybrid and word-only recognizers have different coverage levels.

recognition may perform better.

5.4 Controlling the Number of Graphones

For the full $L = 4$ graphone model, 19k unique graphones are needed to segment all words in the corpus into graphones. These 19k graphones are added to the speech recognizer's lexicon, thus significantly increasing the size of the lexicon. In fact, for the coverage levels explored in the lecture experiments (88% - 99%), the majority of the lexicon is actually graphones. This leads to the question of whether we can trim the graphone set and still preserve performance for open-vocabulary recognition. The graphone set trimming process is described in Section 4.4.

For this experiment, 96% coverage hybrid recognizers are built with trimmed graphone sets of 4k and 2k. These recognizers are then evaluated on the test set. The

	Full Graphone	4k Graphone	2k Graphone	Word-Only
WER	32.8%	33.6%	34.8%	36.8%
SER	73.5%	73.6%	74.5%	75.1%
LER	18.2%	18.6%	19.1%	20.7%

Table 5.4: Comparison of recognition performance for the full graphone hybrid model and trimmed graphone models (4k and 2k graphone set size). Results are also shown for the word-only recognizer. All of these recognizers have 96% vocabulary coverage on the training corpus.

results of these recognition experiments are shown in Table 5.4. The speech recognition performance only degrades slightly when the graphone set is trimmed down from 19k to 4k. The difference between these two setups are less than 1 percentage point for all three metrics. Therefore, the complexity of the recognizer can be significantly reduced while maintaining a similar level of recognition performance.

5.5 Hybrid Recognition with Full Vocabulary

Choosing a vocabulary coverage for the hybrid model has a trade-off. For higher vocabulary coverage, more words are in-vocabulary, but there is also less graphone-to-graphone transitions available in the training corpus for learning. This can reduce the graphones' ability to spell OOV words. For lower vocabulary coverage, although there are more graphone-graphone transitions available, a smaller vocabulary decreases overall performance. The goal of this section is to build a recognizer that has 100% vocabulary coverage of the training corpus, but also has hybrid model knowledge to help it perform even better.

In order to do this, the recognizer must learn N-grams of all words in the training corpus, as well as N-grams containing graphones. We can accomplish this by training a hybrid recognizer on the concatenation of two copies of the language model corpus. The first copy is the original corpus, and the second copy is a hybrid corpus at a chosen vocabulary coverage. The first copy gives us full vocabulary coverage, and the second copy incorporates word-graphone and graphone-graphone transitions into the language model.

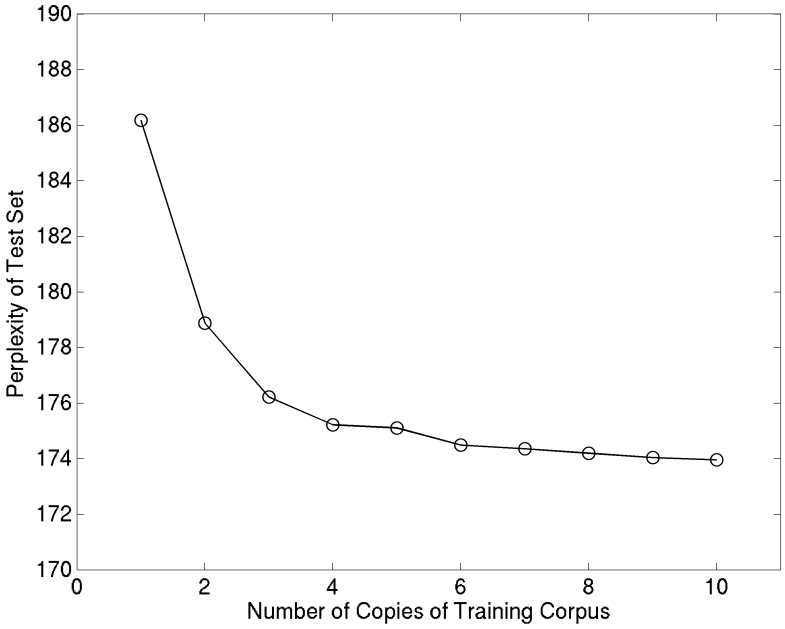

Figure 5-4: Test set perplexity of language models trained on a replicated training corpus.

However, care must be taken when concatenating a language model corpus with itself. Depending on the effects of smoothing, the language model estimated on the same text replicated twice or more times can lead to a different, and perhaps better model. For example, if a graphone model trained on two copies of the training corpus (1 word-only, 1 hybrid) does better than the word-only recognizer, then we are still not sure if the improvement is due to having some sentences appear twice, or from the addition of graphones. In order to have a fair comparison with the proposed graphone model, we must find a point where concatenating additional copies of the language model corpus no longer increases the quality of the language model. To do this, language models are trained on replicated word-only lecture corpora of varying number of copies. Then, their perplexities on the lecture test set are calculated using SRILM. Figure 5-4 shows the result of this perplexity calculation. Because the perplexity curve flattens as we get to 10 copies of the training corpus, we use this configuration to test the full-vocabulary graphone model. For practical applications, these perplexity calculations can be done on a development set.

	Word-only 1x	Word-only 10x	96% Hybrid		98% Hybrid		99% Hybrid	
			OOV	Gr	OOV	Gr	OOV	Gr
WER	31.1%	29.0%	29.3%	29.3%	29.2%	29.2%	29.1%	29.1%
SER	72.0%	69.7%	70.1%	70.1%	69.9%	69.9%	69.8%	69.8%
LER	17.6%	16.5%	-	16.5%	-	16.5%	-	16.5%

Table 5.5: Performance of hybrid recognizers with full vocabulary compared to the full-coverage baseline word-only recognizers. Word-only 1x and 10x are built with one and ten copies of the training corpus, respectively. The OOV column is for replacing graphone sequences by OOV tags. The Gr column is for concatenating graphones together to form words.

For this experiment, nine copies of the original corpora are concatenated with one copy of the hybrid corpus at a given coverage level (96%, 98%, or 99%). The 4k graphone model ($L = 4$) as described in Section 5.4 is used for graphonizing OOV words in the training corpus. This trimmed graphone set is used because these graphones are added to an already-huge dictionary of 49.3k words. The hybrid recognizer built on the resulting training corpus has 100% training vocabulary coverage, plus knowledge about graphones. The hybrid results are compared to a word-only recognizer trained on 10 copies of the original corpora. Table 5.5 shows results of this experiment. For reference, this table also includes the results of the word-only recognizer trained on one copy of the training corpus. These results show that as predicted from perplexity calculations, replicating the training corpus does increase performance, but this is likely to be a side-effect of smoothing. Comparing the 10x word-only language model and corresponding hybrid recognizers (1x hybrid + 9x word-only corpus), we do not see any gains with the hybrid recognizers. We also do not see much decrease in recognition either, especially in LER. Comparing the columns for replacing by an OOV tag and for concatenating graphones, we see that at such a high coverage level, the graphones rarely spell OOV words correctly. Because the training set's coverage of test set words is very high for this task, it is also good too see that adding graphones to a full-coverage recognizer does not significantly degrade recognition. If this full-vocabulary hybrid were built for a different task where the training corpus has a much lower coverage of the test corpus, then the full-vocabulary hybrid has much greater potential for outperforming the full-coverage word-only recognizer.

5.6 Hierarchical Alternative for Hybrid Model

So far, when we mention hybrid models, we have been referring to flat hybrid models. However, there are alternatives to the flat approach for building a hybrid model. In this section, we experiment with *hierarchical* hybrid models, or simply hierarchical models for short. These models are hierarchical because they model the language corpus at two levels: the word level and the subword level. At the word-level, OOV words are replaced by OOV tokens during language model training. The subword-level model is trained on an arbitrary pronunciation dictionary, and serves as the OOV model. During recognition, the top-level recognizer decides whether an OOV word has occurred, and if so, the recognizer uses the subword OOV model to find the most-likely OOV word (see Figure 5-5). A cost factor, C_{oov}, is applied whenever the recognizer uses the OOV model, which can be seen as a normalizing factor between the OOV model and the word-level model. This cost is additive in the log-probability domain, so it is multiplicative in the probability domain. Bazzi and Glass have used phonemes for the OOV model in the hierarchical framework to generate phonetic sequences for OOV words [4]. We build on previous work by using graphones in the OOV model to spell OOV words.

There are pros and cons of using a hierarchical model versus a flat model to detect and hypothesize OOVs. The flat model knows about transitions from words to the beginning graphone of an OOV, as well as the transition from the ending graphone of an OOV to the next word. When there are consecutive OOVs in the training corpus, the flat model also learns transitions between graphones at the boundary of the OOVs. The hierarchical model may have better generalizability for spelling OOV words because its OOV model can be trained on a large dictionary that is independent of the language model training corpus for the recognizer. The flat hybrid on the other hand, only learns how to spell OOV words from the segmented words in the language model training corpus. At high vocabulary coverages, this is usually a much smaller data set than a large training dictionary. The hierarchical model can capture probabilities that span words on both sides of the OOV token, but the flat model

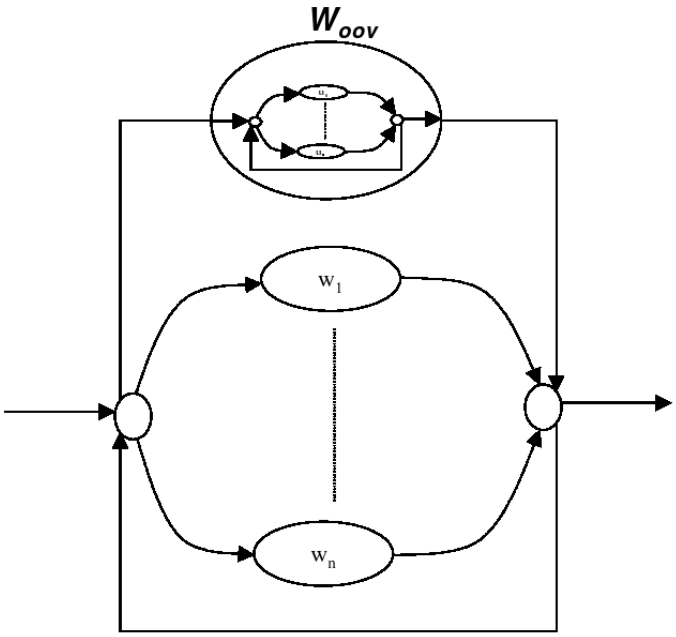

Figure 5-5: Hierarchical OOV model framework (from [2]). W_{OOV} represents the OOV token.

cannot, unless the OOV word is very short. The flat model doesn't require parameter tuning, while the flat hybrid model requires tuning of C_{oov} on a development set. However, tuning this parameter also gives us more control of how often to go into the OOV model, which can help us prevent in-vocabulary words from being recognized as OOVs.

To build the hierarchical model for this experiment, a graphone model is trained on the SLS pronunciation dictionary with parameters $L = 3$, $N = 3$, resulting in a set of 19.6k graphones. This model is then converted to an FST using the MIT Finite-State Transducer Toolkit [26]. The C_{oov} factor is added by inserting a node at the beginning of this FST. For the word-level model, the language model is estimated using SRILM at 98% training set vocabulary coverage, replacing OOV words by an OOV token. This language model is then converted to an FST where the OOV token is converted to a dynamic class that points to the OOV model's FST. A 400-utterance development set is used to tune C_{oov} by optimizing on WER. The best C_{oov} found for this model is -7.5 (FST scores are negative-log probabilities). This negative cost

is needed because a hypothesized word generated by the OOV model has a much higher score than a normal word. The word-transition-weight for the hierarchical recognizer is also adjusted slightly to compensate for the addition of the OOV model. Performing a backward pass on the top-level model is tricky because we need to skip over all the OOV tokens. For this proof-of-concept experiment, we run a forward trigram and widen the search beam by a factor of 5 to compensate for the much larger search space. We plan to implement the bigram forward pass and a trigram backward pass for this model in the near future.

For evaluation, the hierarchical and flat hybrids, along with a word-only recognizer on a 7k test set. The flat hybrid uses a $L = 3$, $N = 5$ L2G converter for graphonizing OOV words in the training set. All of these recognizers have a vocabulary coverage of 98% on the training set. The recognition results are shown Table 5.6. These results show that the hierarchical and flat hybrid models have similar performance, and both significantly outperform the word-only recognizer at this coverage level. Several examples of recognition output for the three types of recognizers are shown in Table 5.7. These examples show that although the flat hybrid model and the top-level N-gram of the hierarchical model are trained on corpora with the same OOV locations, the recognition output can still vary in where OOV words are detected. The hypothesized spellings are often different for the two hybrid models because the training data for the graphone part of these models are different.

Although the LER of the hierarchical model is slightly worse than the hybrid recognizer, a closer examination of the recognizers' output actually reveals that the hierarchical model generalizes better to truly unseen data. There are 204 unique words in the test set that occur nowhere in the training corpus. Out of those unseen words, the flat hybrid spells 10 of those words correctly (including placing the word boundaries correctly) at least once in the test set. The hierarchical hybrid however, spells 41 of those truly unseen words correctly at least once in the test set. This shows that if the OOV position is hypothesized correctly, then the hierarchical hybrid is more capable of spelling the OOV word. This is expected because the hierarchical OOV model is trained on a 150k-word dictionary rather than only the words that

	Word-only	Flat	Hierarchical
WER	33.6%	32.5%	32.5%
SER	73.2%	72.8%	72.8%
LER	19.0%	18.2%	18.3%

Table 5.6: Comparison of hybrid approaches for the lecture transcription task at 98% vocabulary coverage.

get graphonized at the 98% coverage level on the training corpus. In addition, EM with smoothing is run for all N-gram sizes when building the graphone model for the hierarchical hybrid, which is not the case for graphone N-grams in the flat hybrid. Therefore, this confirms one strength of hierarchical OOV models over flat hybrid models.

5.7 Summary

In this chapter, we built a hybrid recognizer for the lectures domain and demonstrated significant improvements over the word-only baseline recognizer. We also showed that trimming the set down to 4k graphones does not significantly impact performance. A full-coverage hybrid recognizer was also built and shown to produce less errors than the full-coverage word-only recognizer. Finally, we presented a hierarchical graphone model that yields similar performance as the flat hybrid graphone model.

Hierarchical	Flat	Word-only	Reference
<uh> represents [cor|bel|hyd|ra|tes]	<uh> represents [car|bo|hyd|ra|tes]	<uh> represents car behind rates	<uh> represents carbohydrates
summary tell me [asy|mpt|ot|ica|lly]	summary tell me [asy|mpt|ot|ica|lly]	summary tell me hasn't ironically	somebody tell me asymptotically
here's the [gol|gi|ap|br|ede|son] the gold yet [app|ara|tus] up here	here's the goal g [app|ara|tus] in the gold yet [app|ara|tus] up here	here's the goal of yep bread so the gold yet <uh> bread is up here	here's the golgi apparatus and the golgi <uh> apparatus up here
this so late of the triangle this the [hyp|ot|enu|se] have flow a and	this so late of the triangle this the hype understand flow a and	this so late of the triangle this the hype understand flow a and	this side of the triangle this the hypotenuse has slope an
and the truth things that vast numbers of [bio|che|mic|a|lly] catches are made by [esp|iri|fic|ati|on] reactions and reversed by reactions that are called simply [hyd|r|oly|sis]	and the truth things that vast numbers of biochemical [ink|a|ges] are made by us [ver|ifi|cat|ion] reactions and reversed by reactions that are called simply [hyd|ro|sis]	and the truth things that vast numbers of biochemical the teachers are made by a spirit vacation reactions and reversed by reactions that are called simply high gross is	and the truth is that vast numbers of biochemical linkages are made by esterification reactions and reversed by reactions that are called simply hydrolysis

Table 5.7: Examples of recognition output from hierarchical, flat, and word-only models. All recognizers have a 98% training corpus coverage. Square brackets denote concatenated graphones (only letter parts shown). Vertical bars show boundaries between graphones.

Chapter 6

Summary and Future Directions

In this thesis, we explored using graphones for letter-to-sound (L2S) as well as hybrid recognition tasks, in several applications with practical use cases. We discussed L2S conversion in context of generating restaurant and street names for a mobile restaurant guide application, and produced results slightly better than similar subword approaches. We then presented results of using graphone flat hybrid recognition to recognize lyrics queries issued by users with the goal of finding songs. Finally, we applied graphones to transcribing spoken lectures in an open vocabulary setting. Overall, we conclude that the flat hybrid model is effective when compared to word-only models with the same vocabulary coverage. We have also seen that trimming the graphone set does not significantly degrade recognition performance, especially for the open vocabulary task. Finally we explored some improvements to the standard flat hybrid model by incorporating all the words in the vocabulary, as well as building a hierarchical hybrid instead of a flat hybrid.

6.1 Future Directions

One future area of focus is improving the flat hybrid model. We can add a word boundary token to graphonized OOVs in the language model training corpus so that we can easily separate consecutive OOV words in the recognition output. This can be especially useful if the OOV rate is high. Also, the graphone size parameters

has trade-offs for hybrid recognition performance because large graphones are good for recognition, but are not as good as smaller graphones for graphonizing words accurately. However this trade-off may be mitigated by using a long N-gram L2S to get pronunciations for OOV words, and then jointly segmenting the OOV words into graphones [39]. Although Vertanen does not find improvements for using this technique on the WSJ task, the performance of this technique might be domain dependent. For future work, we can explore this technique for the lecture and lyrics domains.

We also presented a hierarchical graphone model in this work, but there is much more to be studied in this area. We can dynamically adapt OOV models for the hierarchical hybrid recognizer to improve its ability to hypothesize OOV spellings. The hierarchical model can also be combined with a flat hybrid approach through a two-stage configuration. In this system, the first stage uses a flat hybrid model to identify OOV regions. The second stage rescores these regions with a more powerful graphone-only OOV model similar to the one used for the hierarchical hybrid.

Although building a hybrid model allows us to detect OOVs, we also lose training corpus words from the vocabulary when we decide on a vocabulary coverage for the hybrid model (applies to both flat and hierarchical models). The words excluded from the hybrid model could have been added to the speech recognizer's vocabulary after L2S conversion. The best solution should have the best of both worlds. It should allow the recognizer to use pronunciations of all words in the training corpus, plus information offered by a hybrid model. In this work, we explored a preliminary approach, which is to simply concatenate the word-only corpus with a hybrid corpus for LM training. However, more advanced techniques can be used. One approach could be to merge a large-vocabulary word-only model and a flat-hybrid model through language modeling techniques. Another approach could consider every word in the training corpus to have a some partial probability of being considered as OOV, so that we can effectively learn word-to-word and word-to-graphone transitions for all words in the training corpus.

When the graphones are trained by maximum-likelihood training, we have seen

that the L2S models with larger graphones do not perform as well as the ones with tiny graphones. Ideally, the model with bigger graphones should contain the model with the smaller graphones. If the best model were the model with the smaller graphones, then the ideal training process should always produce the smaller-graphone model, regardless of the specified graphone size upper limit. In the current training process, the smaller graphones are able to take significant advantage of language modeling techniques, but bigger graphones don't get as much benefit because each word does not contain as many big graphones. One possibility is to introduce some form of backoff and smoothing along the graphone size dimension in a manner analogous to the existing backoff and smoothing along the N-gram size dimension. Another possibility is to use a prior based on linguistic knowledge and turn the maximum-likelihood training into a maximum a posteriori training process. This will be helpful for languages where we have some linguistic knowledge about syllable size or structure.

Appendix A

Data Tables: Restaurant and Street Name Recognition

		Top 1 Pronunciation			Top 2 Pronunciations		
L	N	All	Seen	Unseen	All	Seen	Unseen
1	1	65.0	65.5	64.8	65.5	65.8	65.4
1	2	47.2	47.3	47.1	42.9	43.3	42.8
1	3	38.9	41.2	37.8	35.2	36.6	34.6
1	4	38.2	38.7	38.0	33.1	31.8	33.7
1	5	36.0	36.6	35.7	32.5	31.0	33.2
1	6	35.2	35.8	34.9	31.6	30.0	32.4
1	7	35.6	35.5	35.7	32.0	30.7	32.6
1	8	35.6	35.5	35.7	32.0	30.8	32.6
2	1	58.3	60.2	57.5	55.0	55.9	54.5
2	2	38.8	39.8	38.4	33.5	33.5	33.5
2	3	37.4	37.2	37.6	33.1	31.0	34.1
2	4	37.0	36.6	37.3	33.1	32.4	33.5
2	5	36.8	36.4	37.0	32.9	32.4	33.2
2	6	36.8	36.4	37.0	32.9	32.4	33.2
3	1	51.7	51.8	51.7	45.5	45.7	45.4
3	2	37.5	36.6	37.9	34.0	34.0	34.0
3	3	36.6	35.6	37.0	33.8	32.9	34.3
3	4	36.6	35.6	37.0	33.8	32.9	34.3
3	5	36.6	35.6	37.0	33.8	32.9	34.3
4	1	44.4	43.0	45.1	40.2	39.6	40.5
4	2	37.8	36.4	38.4	34.5	32.9	35.2
4	3	37.6	36.1	38.2	34.4	32.7	35.1
4	4	37.5	36.1	38.1	34.4	32.7	35.1
4	5	37.5	36.1	38.1	34.4	32.7	35.1

Table A.1: Word recognition performance using automatically-generated pronunciations for restaurant and street names. Graphone L2S converters with various L and N values are used to generate pronunciations. Results are shown in WER (%). This data is used to generate Figures 3-1, 3-2, 3-4, 3-5, 3-6, and 3-7.

Num. Pron.	Graphone	Spellneme
1	35.2	37.1
2	31.6	32.1
3	30.6	30.6
5	30.3	30.2
10	30.0	29.7
20	30.3	30.4
50	32.3	31.1

Table A.2: Word recognition performance with various number of pronunciations. The graphone converter is trained with $L = 1$, and $N = 6$. The spellneme data is from [13]. Results are shown in WER (%). This data corresponds to Figure 3-3.

Appendix B

Data Tables: Lyrics Hybrid Recognition

			Word-only			Word & OOV		Word & Graphone		
C_{train}	C_{test}	$\|V\|$	WER	SER	LER	WER	SER	WER	SER	LER
50.0	53.0	68	88.1	99.3	55.5	63.4	99.4	48.6	89.5	23.9
60.0	63.5	120	79.4	97.5	51.2	55.9	98.0	41.8	84.3	21.9
70.0	74.3	233	65.3	93.7	41.9	47.3	94.3	35.8	77.9	20.1
80.0	84.5	492	50.1	86.0	32.6	39.1	87.3	31.6	72.7	18.8
90.0	93.4	1443	36.9	75.7	23.9	31.7	76.2	28.5	67.9	17.6
92.0	94.9	1942	34.8	73.3	22.3	30.5	74.3	28.0	67.4	17.4
94.0	96.4	2766	32.5	71.1	21.1	29.2	70.9	27.5	66.8	17.2
96.0	97.7	4386	30.5	68.9	19.9	28.0	68.9	27.1	66.2	17.0
98.0	99.0	8572	29.2	67.5	19.0	27.3	66.4	27.0	65.4	16.9
99.0	99.4	14532	28.6	67.5	18.6	26.6	65.8	26.4	65.5	16.6
100.0	99.8	46937	28.4	67.5	18.5	-	-	-	-	-

Table B.1: Performance of the flat hybrid recognizer on music lyrics. C_{train} and C_{test} are vocabulary coverages (%) on the training and test sets, respectively. $|V|$ is the vocabulary size. Word & OOV is replacing each graphone sequence in the recognizer's output by an OOV tag. Word & Graphone is concatenating graphones to spell OOV words. All error rates are in percentages. This data corresponds to Figures 4-1, 4-2, and 4-3.

Appendix C

Data Tables: Spoken Lecture Transcription

			Word-only			Word & OOV		Word & Graphone		
C_{train}	C_{test}	$\|V\|$	WER	SER	LER	WER	SER	WER	SER	LER
88.0	85.6	1320	52.5	80.3	29.9	41.9	80.7	36.2	75.5	19.3
90.0	88.4	1780	47.5	78.6	26.9	39.8	79.1	35.0	74.5	18.9
92.0	90.4	2468	43.9	77.7	24.7	38.2	78.2	34.3	74.0	18.7
94.0	93.0	3594	39.6	76.3	22.1	36.3	76.7	33.6	73.8	18.4
96.0	95.0	5661	36.8	75.1	20.7	34.7	75.5	32.8	73.5	18.2
98.0	97.2	10634	36.6	73.4	18.9	33.0	73.8	32.2	73.0	18.0
99.0	98.4	17374	32.1	72.5	18.1	32.1	72.8	31.8	72.7	17.8
100.0	99.4	49333	31.1	72.0	17.6	-	-	-	-	-

Table C.1: Performance of the flat hybrid recognizer on transcribing spoken lectures. C_{train} and C_{test} are vocabulary coverages (%) on training and test sets, respectively. $|V|$ is the vocabulary size. Word & OOV is replacing each graphone sequence in the recognizer's output by an OOV tag. Word & Graphone is concatenating graphones to spell OOV words. All error rates are in percentages. This data corresponds to Figures 5-1, 5-2, and 5-3.

Bibliography

[1] M. Akbacak, D. Vergyri, and A. Stolcke. Open-vocabulary spoken term detection using graphone-based hybrid recognition systems. In *Proc. Int. Conf. on Spoken Language Processing*, pages 5240–5243, Las Vegas, NV, USA, 2008.

[2] I. Bazzi. *Modelling out-of-vocabulary words for robust speech recognition*. Ph.D. Thesis. MIT Department of Electrical Engineering and Computer Science, 2002.

[3] I. Bazzi and J. Glass. Heterogeneous lexical units for automatic speech recognition: Preliminary investigations. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, 2000.

[4] I. Bazzi and J. Glass. Modeling out-of-vocabulary words for robust speech recognition. In *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, 2000.

[5] I. Bazzi and J. Glass. Learning units for domain-independent out-of-vocabulary word modelling. In *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, 2001.

[6] M. Bisani and H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. Int. Conf. on Spoken Language Processing*, pages 105–108, Denver, CO, USA, 2002.

[7] M. Bisani and H. Ney. Multigram-based grapheme-tophoneme conversion for lvcsr. In *Proc. European Conf. on Speech Communication and Technology*, pages 933–936, Geneva, Switzerland, 2003.

[8] M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Proc. European Conf. on Speech Communication and Technology*, pages 725–728, Lisbon, Portugal, 2005.

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

[10] L. Burget, P. Schwarz, P. Matějka, M. Hannemann, A. Rastrow, C. White, S. Khudanpur, H. Hermansky, and J. Černocký. Combination of strongly and weekly constrained recognizers for reliable detection of OOVs. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4081–4084, Las Vegas, NV, USA, 2008.

[11] S. F. Chen. Conditional and joing models for grapheme-to-phoneme converstion. In *Proc. European Conf. on Speech Communication and Technology*, pages 2033–2036, Geneva, Switzerland, 2003.

[12] G. Choueiter. *Linguistically-Motivated Sub-word Modeling with Applications to Speech Recognition*. Ph.D. Thesis. MIT Department of Electrical Engineering and Computer Science, 2009.

[13] G. Choueiter, S. Seneff, and J. Glass. Automatic lexical pronunciations generation and update. In *Proc. Workshop on Automatic Speech Recognition and Understanding*, pages 225–230, Kyoto, Japan, 2007.

[14] G. Chung. A three-stage solution for flexible vocabulary speech understanding. In *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, 2000.

[15] W. Daelemans and A. van den Bosch. Language-independent data-oriented grapheme-to-phoneme conversion. In *Progress in Speech Processing*, pages 77–89, 1997.

[16] S. Deligne and F. Bimbot. Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23:223–247, 1997.

[17] S. Deligne, F. Yvon, and F. Bimbot. Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc. European Conf. on Speech Communication and Technology*, pages 2243–2246, Madrid, Spain, 1995.

[18] H. Elovitz, R. Johnson, A. McHugh, and J. Shore. Letter-to-sound rules for automatic translation of english text to phonetics. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 24:446–459, 1976.

[19] L. Galescu. Recognition of out-of-vocabulary words with sub-lexical language models. In *European Conf. on Speech Communication and Technology*, pages 249–252, Geneva, Switzerland, 2003.

[20] M. Gerosa and M. Federico. Coping with out-fo-vocabulary words: open vursus huge vocabulary asr. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4313–4316, Taipei, Taiwan, 2009.

[21] J. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17(2-3):137–152, 2003.

[22] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. Recent progress in the MIT spoken lecture processing project. In *Proc. Interspeech*, pages 2553–2556, Antwerp, Belgium, 2007.

[23] T. Hazen and I. Bazzi. A comparison and combination of methods for OOV word detection and word confidence scoring. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, USA, 2001.

[24] T. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Recognition confidence scoring for use in speech understanding systems. In *Computer Speech and Language*, pages 49–67, 2000.

[25] I. Hetherington. *A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding.* Ph.D. Thesis. MIT Department of Electrical Engineering and Computer Science, 1995.

[26] L. Hetherington. The MIT finite-state transducer toolkit for speech and language processing. In *Int. Conf on Spoken Language Processing*, Jeju, South Korea, 2004.

[27] S. Jiampojamarn, C. Cherry, and G. Kondrak. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proc. Workshop on Human Language Technology*, pages 905–913, Columbus, OH, USA, 2008.

[28] T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *Proc. European Conf. on Speech Communication and Technology*, pages 827–830, Rhodes, Greece, 1997.

[29] H. Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff. OOV detection by joint word/phone lattice alignment. In *Proc. Workshop on Automatic Speech Recognition and Understanding*, pages 478–483, 2007.

[30] C. Martins, A. Teixeira, and J. Neto. Vocabulary selection for a broadcast news transcription system using morpho-syntatic approach. In *Proc. Int. Conf. on Spoken Language Processing*, Antwerp, Belgium, 2007.

[31] H. M. Meng, S. Seneff, and V. W. Zue. Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation. In *Proc. Workshop on Human Language Technology*, pages 289–294, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[32] S. Oger, G. Linarès, F. Béchet, and P. Nocera. On-demand new word learning using world wide web. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4305–4308, Las Vegas, NV, USA, 2008.

[33] A. Rastrow, A. Sethy, and B. Ramabhadran. A new method for OOV detection using hybrid word/fragment system. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 3953–3956, Taipei, Taiwan, 2009.

[34] S. Seneff. The use of subword linguistic modeling for multiple tasks in speech recognition. *Speech Communication*, 42(3-4):373–390, 2004.

[35] S. Seneff. Reversible sound-to-letter/letter-to-sound modeling based on syllabic structure. In *Proc. Workshop on Human Language Technology*, pages 153–156, Rochester, NY, 2007.

[36] A. Stolcke. SRILM - An extensible language modeling toolkit. In *Proc. Int. Conf on Spoken Language Processing*, Denver, CO, USA, 2002.

[37] P. Taylor. Hidden markov models for grapheme to phoneme conversion. In *Proc. European Conf. on Speech Communication and Technology*, Lisbon, Portugal, 2005.

[38] K. Torkkola. An efficient way to learn english grapheme-to-phoneme rules automatically. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 199–202, Martigny, Switzerland, 1993.

[39] K. Vertanen. Combining open vocabulary recognition and word confusion networks. In *Proc. Int. Conf. on Spoken Language Processing*, pages 4325–4328, Las Vegas, NV, USA, 2008.

[40] P. Vozila, J. Adams, Y. Lobacheva, and R. Thomas. Grapheme to phoneme conversion and dictionary verification using graphonemes. In *Proc. European Conf. on Speech Communication and Technology*, pages 2469–2472, Geneva, Switzerland, 2003.

[41] F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

[42] C. White, G. Zweig, L. Burget, P. Schwarz, and H. Hermansky. Confidence estimation, OOV detection and language ID using phone-to-word transduction and phone-level alignment. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4085–4088, Las Vegas, NV, USA, 2008.

[43] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington. JUPITER: A telephone-based conversational interface for weather information. *IEEE Trans. on Speech and Audio Processing*, 8(1):100–112, 2000.