Learning Lexicons From Speech Using a Pronunciation Mixture Model

Ian McGraw, Ibrahim Badr, and James R. Glass, *Senior Member, IEEE*

***Abstract*—In many ways, the lexicon remains the Achilles heel of modern automatic speech recognizers. Unlike stochastic acoustic and language models that learn the values of their parameters from training data, the baseform pronunciations of words in a recognizer's lexicon are typically specified manually, and do not change, unless they are edited by an expert. Our work presents a novel generative framework that uses speech data to learn stochastic lexicons, thereby taking a step towards alleviating the need for manual intervention and automatically learning high-quality pronunciations for words. We test our model on continuous speech in a weather information domain. In our experiments, we see significant improvements over a manually specified "expert-pronunciation" lexicon. We then analyze variations of the parameter settings used to achieve these gains.**

***Index Terms*—Baseform generation, dictionary training with acoustics via EM, pronunciation learning, stochastic lexicon.**

I. Introduction

WITHOUT question, automatic speech recognition is a data-driven technology. It is disconcerting, then, that the list of word-pronunciations found in the lexicon, a central component of almost any speech recognizer, is typically static and manually updated rather than learned probabilistically. For large vocabulary speech recognition, the research community often relegates the modeling of phonological variation to context-dependent acoustic models. By contrast, this work explores the loosening of phonetic constraints at the lexical level with the help of a straightfoward application of Expectation-Maximization (EM) to learning a stochastic lexicon. Central to this formulation is a shift in perspective regarding the objective of the lexicon itself. Rather than providing a mapping between each word and one, or perhaps a few, canonical pronunciations, the stochastic lexicons trained in this work theoretically model a weighted mixture of all possible phonetic realizations of a word. This leads us to refer to these stochastic lexicons as pronunciation mixture models (PMMs).

The work presented here is an extension of our previous exploration of the PMM framework [1] and [2]. These papers explore a maximum likelihood training procedure for the PMM that incorporates information from spoken examples of a word, in addition to its spelling, into automatically learned pronunciation weights. In [1], we simulate missing pronunciations on an isolated word corpus, and use the PMM to recover expert-quality pronunciations. In [2], we extend this framework to continuous speech and show that, even with well-matched acoustic models, the PMM improves recognition performance. In both cases, we make heavy use of a state-of-the-art letter-to-sound (L2S) system based on joint-sequence modeling [3].

L2S systems are often used when hand-crafted pronunciations fail to cover the vocabulary of a particular domain. Often the parameters of these models are trained only using existing lexicons. The pronunciation mixture model provides a principled approach of incorporating acoustic information into L2S pronunciation generation. Moreover, rather than limiting the pronunciation generation to out-of-vocabulary words, the PMM can effectively be applied to training the entire lexicon. Like the acoustic and language models, the data used to learn pronunciations would ideally match the test domain as closely as possible. Indeed, the experiments of this work use the very same training data. Since this data is readily available at no extra cost for most recognizers, we hope these results encourage the adoption of lexicon learning as a standard phase in training an automatic speech recognition (ASR) system.

More broadly, we view this work as a small step towards being able to train a speech recognizer entirely from an orthographically transcribed corpus. Ideally, the lexical pronunciations and perhaps even the basic phonetic units of a language themselves could be determined automatically from a large amount of transcribed speech data. Were such problems solved, the process of training a speech recognizer might be reduced to a black-box procedure for which even a non-expert could input the necessary training data, and from which would emerge a speech recognizer of the appropriate language. For some languages, this vision is already close at hand, while for others there are significant problems yet to be solved. English, in particular, presents difficulties due to the irregular mapping between letters and sounds.

This motivates us to concentrate this work on the issue of pronunciation generation while, for the moment, leaving the existing linguistically inspired phonetic units largely intact. In Section II, background information is presented which reformulates the basic equations governing modern speech recognition technology with an additional term for the stochastic lexicon. Section III then details how the recognizer search space is implemented using a weighted finite state tranducer (FST) landmark-based recognizer. Sections IV and V provide a review of the literature related to this work followed by a more detailed overview of the L2S framework used in the majority of our ex-

Manuscript received January 23, 2012; revised June 16, 2012 and October 16, 2012; accepted October 18, 2012. Date of publication October 23, 2012; date of current version December 10, 2012. This work was supported in part by the Qmulus Project, a joint research program between MIT and Quanta Computer Inc. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Steve Renals.

The authors are with the Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: imcgraw@csail.mit.edu; iab02@csail.mit.edu; glass@mit.edu).

Digital Object Identifier 10.1109/TASL.2012.2226158

periments. We then review a general formulation for the PMM in Section VI, and discuss issues that arise when implementing the PMM in practice in Section VII. Section VIII then extends the experiments of [2] on a weather query corpus with additional analysis. Finally, we briefly discuss the phonological characteristics of the learned stochastic lexicons in Section IX, before concluding with a brief summary in Section X.

II. Background

The problem of ASR is typically formulated as the search for a series of words through a distribution of hypotheses modeled statistically. Conceptually, the search space is layered. Individual words are composed of phonetic units, which are in turn modeled using states in probabilistic machinery such as a finite state transducer (FST). Across each layer, constraints are introduced. A language model captures the likelihood of word sequences, perhaps using a variation of the standard N-gram approach. At a lower level, a context-dependent acoustic model bundles together the phonetic units. In this work, we are concerned mainly with the relation between two layers: the words and the phone sequences that constitute their pronunciations. For example, the word *colonel* might be mapped to a pronunciation by an expert, and end up in the lexicon in an *Arpabet* representation as *colonel: k er n ax l*.

Automatic speech recognition is often motivated with a few short equations. In particular, the goal of ASR is to find the most likely sequence of words $\mathbf{W}^* = \mathbf{w}_1^*, \ldots, \mathbf{w}_n^*$ given an utterance $\mathbf{u}$. To do this, we define a distribution P to be our search space, and model the probability of a word sequence given the acoustic signal as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{u}) \quad (1)$$

$$= \arg\max_{\mathbf{W}} P(\mathbf{u}|\mathbf{W})P(\mathbf{W}) \quad (2)$$

We let $P(\mathbf{W})$ represent the language model and $P(\mathbf{u}|\mathbf{W})$ represent the probability of the acoustics given a particular transcript. Assuming a deterministic lexicon, in which a word is paired with a single pronunciation, the uncertainty in (2) is entirely relegated to the language and acoustic models, and the lexicon serves to couple the two with tight constraints. It is more informative and less restrictive, however, to explicitly view the lexicon as its own statistical component. The following equations describe how one might treat the pronunciations underlying a word sequence as a hidden variable:

$$P(\mathbf{W}|\mathbf{u}) = \sum_{\mathbf{B}\in\mathcal{B}_\#} P(\mathbf{W},\mathbf{B}|\mathbf{u}) \quad (3)$$

$$= \frac{\sum_{\mathbf{B}\in\mathcal{B}_\#} P(\mathbf{W},\mathbf{B},\mathbf{u})}{P(\mathbf{u})} \quad (4)$$

In this work, we denote the set of all possible sequences of phonological units using $\mathcal{B}$, and where word-boundary delimiters are included we use $\mathcal{B}_\#$. Thus, a particular pronunciation sequence is specified by $\mathbf{B} = \mathbf{b}_1\#\mathbf{b}_2\ldots\#\mathbf{b}_K \in \mathcal{B}_\#$, where each $\mathbf{b}_i \in \mathcal{B}$. Factoring the equation above further we have,

$$P(\mathbf{W}|\mathbf{u}) = \frac{\sum_{\mathbf{B}\in\mathcal{B}_\#} P(\mathbf{u}|\mathbf{B})P(\mathbf{B}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{u})} \quad (5)$$

where we make the conditional independence assumption that the acoustics are independent of the words given their pronunciations, $p(\mathbf{u}|\mathbf{B},\mathbf{W}) = P(\mathbf{u}|\mathbf{B})$. With our new $P(\mathbf{W}|\mathbf{u})$ we see that:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}) \sum_{\mathbf{B}\in\mathcal{B}_\#} P(\mathbf{u}|\mathbf{B})P(\mathbf{B}|\mathbf{W}) \quad (6)$$

To ensure that the computation required for decoding is tractable, the *Viterbi* approximation is used, in which the summation over all possible pronunciations is replaced by a maximization.

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \max_{\mathbf{B}} P(\mathbf{u}|\mathbf{B})P(\mathbf{B}|\mathbf{W})P(\mathbf{W}) \quad (7)$$

Modulo a possible penalty for word boundaries we let $P(\mathbf{B}|\mathbf{W}) = P(\mathbf{b}_1|\mathbf{w}_1)\ldots P(\mathbf{b}_n|\mathbf{w}_n)$ represent the way a lexicon contributes to the score of a given hypothesis. This decomposition makes the assumption, common to most lexicons, that a pronunciation of a word is independent of surrounding pronunciations given the word itself. Of course, in typical speech recognizers, lexicons are unweighted, effectively setting $P(\mathbf{b}_i|\mathbf{w}_i) = 1$ for all pronunciations. For words with multiple pronunciations, this does not model a proper probability distribution.

In this work, we argue that it is beneficial to make full use of a lexicon's potential stochasticity. In particular, we will parameterize $P(\mathbf{B}|\mathbf{W})$ and use *Expectation-Maximization* (EM) [4] to find the maximum likelihood parameter estimates given a set of training data. In this way, the lexicon fits nicely into the probabilistic framework other components of an automatic speech recognizer already enjoy [5], and its training is well-motivated through its incorporation into the fundamental equations of recognition described above.

III. The SUMMIT Landmark-Based Speech Recognizer

This work makes use of the SUMMIT speech recognizer [6]. Perhaps the largest difference between SUMMIT and a typical speech recognizer is in its observation space. Rather than computing a sequence of observations at a fixed frame-rate, SUMMIT has been designed to look for places of significant spectral change to identify possible acoustic-phonetic events. These hypothesized *boundary* points form the bases of the recognizer's search space during decoding. Acoustic models are trained on observations from boundaries in the following manner. MFCC averages are taken over varying durations around the landmarks to generate large feature vectors, which are whitened and have their dimensionality reduced through principal components analysis (PCA). Gaussian mixture models are trained for the set of diphone boundary labels found in the training corpus.

Each component of SUMMIT is represented using MIT's open-source finite state transducer (FST) toolkit [7]. Weighted FSTs have the flexibility to represent the variety of the constituent probability distributions of an automatic speech recognizer (e.g. those in (7)). Furthermore, well understood algorithms can be implemented for operations such as composition, denoted with $\circ$, which corresponds roughly to taking a

product of two distributions represented by FSTs. When left unweighted, FSTs can also describe deterministic manipulations of the search space, such as converting between context independent phonetic units and the context dependent boundary units described above.

Another component found in some recognizers that can be efficiently represented using FSTs is a set of phonological rules [8]. Instead of mapping words directly to phone sequences, these rewrite rules are one way that researchers have attempted to account for phonological variation. For example, the phrase "nearest tornado" might be represented at the *phoneme* level as (1) in the table below, but with the application of phonological rules the *phone* pronunciation might read as either (2) or (3).

(1)	n ih r ax s td	t ao r n ey dx ow
(2)	n ih axr s	tcl t er n ey dx ow
(3)	n ih r ax s tcl t	tcl t ao r n ey dcl d ow

Notice the difference in consonant closures and even vowel substitutions. These manually crafted phonological rules account for acoustic-phonetic mismatches between the underlying phonemic baseforms and the surface-level phonetic units, and have been shown to outperform relying on context-dependent acoustic models to implicitly model phonetic variation [9]. Their utility, however, must be re-evaluated in light of stochastic lexicons.

In summary, the SUMMIT FST search space has four primary hierarchical components: the language model (G), the phoneme lexicon (L), the phonological rules (P) that expand the phoneme pronunciations to their phone variations, and the mapping from phone sequences to context-dependent model labels (C). The full network can be represented as a composition of these components: $R = C \circ P \circ L \circ G$. In this work, we will be replacing the hand-crafted P and L with lexicons learned automatically. While we do not entirely remove the dependency on these hand-crafted components, their role is reduced to providing an initialization for a principled training procedure.

IV. Related Work

Research focusing on the speech recognizer's lexicon is often categorized as either addressing pronunciation variation or addressing a version of the out-of-vocabulary (OOV) problem. An early overview of directly modeling phonological variation at the lexical level can be found in [10]. This section summarizes some of this research, as well as more recent endeavors in pronunciation modeling. The particular instantiation of the OOV problem most relevant to this work is the one in which the spelling of the word is known, however, the sequences of phonetic units that comprise its pronunciations are not. To solve this problem, a number of strategies have been devised to construct or train L2S models. In some sense, the PMM connects these two areas of research by combining L2S models with the phonological variation present in acoustic training data. For this reason, we pay particular attention to related work that fits into this middle-space.

We begin, however, by discussing pronunciation generation from the perspective of the OOV problem. Almost all speech recognizers have a finite vocabulary. If the recognizer encounters an out of vocabulary (OOV) term, a word that is not in the lexicon, it cannot produce a transcript that contains it. This complication is compounded in many ASR applications, since a misrecognized OOV word can easily cause the surrounding words to be misrecognized [11]. While one cannot completely eliminate OOVs in an open domain recognition task, techniques have been devised to mitigate the issue, including using confidence scoring to detect OOVs [12], as well as filler models to hypothesize the pronunciation of an OOV word [13]. A particularly common approach to the OOV problem, however, is simply to increase the vocabulary size, whether manually or automatically.

For these reasons, generating pronunciations for new words is the subject of a large body of research [3], [14]–[20]. Although for some languages mapping a spelling to a pronunciation is relatively straightforward, English has shown itself to be rather challenging. Initial work in grapheme-to-phoneme conversion often consisted of rule-based methods [15], however, these quickly ran into issues of scalability and were soon replaced by data-driven methods. A hybrid approach that utilizes linguistic knowledge in a statistical framework is exemplified in [16]. As a first step, a hand-written grammar parses words into a set of linguistically motivated sub-word "spellneme" units. Then, after parsing a large lexicon into these segmented words, an N-gram model is trained and used later for decoding. Alternative L2S approaches include local classification, pronunciation by analogy, and even statistical machine translation [19]. The local classification approach processes a word spelling sequentially from left-to-right and a decision is made for each input character by looking at the letter's context using decision trees [21] or neural networks [22]. Pronunciation by analogy, on the other hand, scans the training lexicon for words or part-of-words that are in some sense similar to the word for which a pronunciation is needed [17], [18]. The output pronunciation is then chosen to be analogous to the existing examples. Finally, the joint-multigram framework of [3] and [23] learns a language model over graphone units, which contain both graphemes and phones. This approach has been shown to produce state-of-the-art results for many L2S tasks. Some have gone on to explore discriminative training in a joint-sequence setting [20].

As evidenced by evaluation metrics, such phoneme error rate, which often assume a canonical baseform, pronunciation generation for OOVs is rarely characterized as an attempt to model pronunciation variation. Still, this variation has been identified as a major cause of errors for a variety of ASR tasks [24], and has therefore received attention from a number of research endeavors grouped into the rather nebulous topic of pronunciation modeling. Of particular interest to this work, are instances in which spoken examples are used to refine pronunciations [25]–[29]. The work of [27], for example, deduces a pronunciation $\mathbf{b}^*$ given a word or grapheme sequence $\mathbf{w}$ and an utterance $\mathbf{u}$ of the spoken word $\mathbf{w}$. This research uses a decision tree to model $P(\mathbf{b}|\mathbf{w})$ which was later shown to produce poor results when compared to graphone models on L2S tasks. The

work of [29] uses the forced alignment of a phonetic decoder to generate a list of possible pronunciations for words, and then assigns weights using a Minimum-Classification-Error criterion. They then test on a business name query corpus. Perhaps the work most similar to our own is that of [28], which makes use of Expectation-Maximization (EM) to adapt graphone language model parameters using acoustic data. Li *et al.* train an initial set of graphone parameters and then adapt them using spoken examples of proper names. They also experiment with discriminative training and show that it produces slightly better results than maximum likelihood estimation (MLE). In our work, rather than adapt the graphone parameters we learn the weights of a stochastic lexicon directly using a similar MLE approach. We then experiment with the PMM on continuous speech data.

V. Joint-Sequence Models (Graphones)

This section reviews the joint-multigram modeling technique of [3], which will later be used to initialize the pronunciation mixture model. We begin with a characterization of the letter-to-sound problem in terms of a joint distribution over grapheme and phone sequences. We let $\mathbf{w}$ denote a particular grapheme sequence in the set of all possible grapheme sequences $\mathcal{W}$ and $\mathbf{b}$ denote a phone sequence drawn from the set of all possible phoneme sequences, $\mathcal{B}$. We then construct the distribution $P(\mathbf{w}, \mathbf{b})$ to represent the probability of the co-occurrence of a particular spelling and pronunciation. The L2S operation that is applied to this joint distribution to recover the optimal pronunciation for a particular word $\hat{\mathbf{w}}$ is the following:

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\hat{\mathbf{w}}, \mathbf{b}) \tag{8}$$

To model $P(\mathbf{w}, \mathbf{b})$, Bisani and Ney capture the relationship between graphemes and phones in a language model over shared units called joint-multigrams. In this work, one or more graphemes are paired with one or more phones to create a new joint-unit called a *graphone*. In previous work, we had also explored the use of *graphonemes* which map graphemes to phonemes; however, here we concentrate mostly on the former. A graphone, $g = (w, b) \in \mathcal{G} \subseteq \mathcal{W} \times \mathcal{B}$, is a sub-word unit that maps a grapheme subsequence, w, to a phone subsequence, b. In previous work, we restricted our attention to *singular* graphones, in which a mapping was made between at most one grapheme and at most one phonetic unit. In general, however, two parameters L and R can be specified, which limit the number of graphemes that appear on the left-hand-side and the number of phones that appear on the right-hand-side of a graphone. While the empty subsequence, ϵ, is allowed, the mapping from ϵ to ϵ is omitted.

Taken together, a sequence of graphones, $\mathbf{g}$, inherently specifies a unique sequence of graphemes $\mathbf{w}$ and phones $\mathbf{b}$; however, the reverse is not the case. There may be multiple ways to align the pair $(\mathbf{w}, \mathbf{b})$ into various graphone sequences $\mathbf{g} \in S(\mathbf{w}, \mathbf{b})$. The following table shows two possible graphone segmentations of the word "couple". In this case, $L = 1$ and $R = 2$.

$\mathbf{w}$	=	c	o	u	p		l	e
$\mathbf{b}$	=	kcl_k	ah		pcl_p	ax	l	
	=	kcl_k		ah	pcl_p	ax	l	
$\mathbf{g}_1$	=	c/kcl_k	o/ah	u/ϵ	p/pcl_p	ϵ/ax	l/l	e/ϵ
$\mathbf{g}_2$	=	c/kcl_k	o/ϵ	u/ah	p/pcl_p	ϵ/ax	l/l	e/ϵ

Given this ambiguity, employing graphonemes in our joint model requires us to marginalize over all possible segmentations. Fortunately, the standard Viterbi approximation has been shown to incur only minor degradation in performance [3].

$$P(\mathbf{w}, \mathbf{b}) = \sum_{\mathbf{g}\in S(\mathbf{w},\mathbf{b})} P(\mathbf{g}) \approx \max_{\mathbf{g}\in S(\mathbf{w},\mathbf{b})} P(\mathbf{g}) \tag{9}$$

The final, albeit rather involved, step is to model $P(\mathbf{g})$ using standard language modeling techniques. The difficulty in training a standard M-gram is that we do not have the necessary training data in the form of graphone sequences. Instead, we are forced to use Expectation-Maximization to generate a set of expected alignments. Our work makes use of an open source implementation of this training procedure, the details of which are described in [3].

VI. The Pronunciation Mixture Model

In Section II, we provided a mathematical foundation of speech recognition that explicitly models the lexicon stochastically. Equation (7) succinctly describes the manner in which such a lexicon contributes to the score of a path during decoding. We now turn our attention to learning the underlying weights of each pronunciation using a pronunciation mixture model. Although initially explored for the isolated word case [1], we subsequently confirmed the PMM's utility on continuous speech data [2]. Here, we present the model for continuous speech, and show that applying it to isolated words is merely a special case.

In stark contrast with models that search for the single most probable pronunciation, the PMM is designed to cope with words that have multiple pronunciations, such as "either". It probably does not make sense, for example, to have one utterance pronounced *iy dh er* and a second pronounced *ay dh er* both vying for a single canonical pronunciation spot. Instead, in our model, both utterances are effectively allowed to distribute soft votes to a mixture of possible pronunciations. Note also that this is an extreme example since the variation occurs at the phoneme level, but as we will show the PMM is able to capture more subtle variation at the phone level as well.

The training data we use to learn the lexicon's weights can be identical to the data used for language and acoustic model training. Suppose, for instance, that it is comprised of M utterances and their transcriptions $D_C = \{\mathbf{u}_i, \mathbf{W}_i\}$ where $\mathbf{u}_i$ is a continuous speech signal and $\mathbf{W}_i = \mathbf{w}_1^i, \cdots, \mathbf{w}_{k_i}^i$ but the word boundary locations within the audio are unknown. We can parameterize the log-likelihood of this data as follows:

$$\mathcal{L}(\theta|D_C) = \sum_{i=1}^{M} \log P(\mathbf{u}_i, \mathbf{W}_i; \theta) \tag{10}$$

$$= \sum_{i=1}^{M} \log \sum_{\mathbf{B}\in\mathcal{B}_\#} P(\mathbf{u}_i, \mathbf{B}, \mathbf{W}_i; \theta) \tag{11}$$

The joint distribution in the previous equation has already been decomposed into the basic components of a speech recognizer in (7). We now incorporate our probabilistic lexicon,

making the assumption that a pronunciation $\mathbf{b}_j$ in an arbitrary pronunciation sequence $\mathbf{B}$ is context independent:

$$P(\mathbf{u}_i, \mathbf{B}, \mathbf{W}_i; \theta) = P(\mathbf{u}_i|\mathbf{B})P(\mathbf{B}|\mathbf{W}_i; \theta)P(\mathbf{W}_i) \quad (12)$$

$$= P(\mathbf{u}_i|\mathbf{B}) \left(\prod_{j=1}^{k_i} P(\mathbf{b}_j|\mathbf{w}_j^i; \theta) \right) P(\mathbf{W}_i) \quad (13)$$

Clearly only the terms associated with the lexicon need be dependent upon its parameters. We now make their use explicit with $\theta_{\mathbf{b}_j|\mathbf{w}_j^i} = P(\mathbf{b}_j|\mathbf{w}_j^i; \theta)$. The goal now, is to maximize the likelihood of the data with respect to these parameters. Note that the language model term vanishes since it does not affect the maximization.

$$\theta^* = \arg\max_\theta \mathcal{L}(\theta|D_C) \quad (14)$$

$$= \arg\max_\theta \sum_{i=1}^{M} \log \sum_{\mathbf{B} \in \mathcal{B}_\#} P(\mathbf{u}_i|\mathbf{B}) \prod_{j=1}^{k_i} \theta_{\mathbf{b}_j|\mathbf{w}_j^i} \quad (15)$$

A local maximum of $\mathcal{L}(\theta|D_C)$ with respect to θ can be obtained using EM. The expectation step computes the expected number of times an alignment between a word $\mathbf{w}$ and pronunciation $\mathbf{p}$ occurs within the data. The maximization step then finds the optimal parameter values given these expected counts. This continues iteratively until convergence to a local maximum.

$$\mathrm{E-step}: \quad \overline{\mathrm{M}}_\theta[\mathbf{w}, \mathbf{p}] = \sum_{i=1}^{M} \sum_{\mathbf{B} \in \mathcal{B}_\#} P(\mathbf{B}|\mathbf{u}_i, \mathbf{W}_i; \theta) \times \mathrm{M}[\mathbf{p}, \mathbf{w}, \mathbf{W}_i, \mathbf{B}] \quad (16)$$

$$\mathrm{M-step}: \quad \theta^*_{\mathbf{p}|\mathbf{w}} = \frac{\overline{\mathrm{M}}_\theta[\mathbf{w}, \mathbf{p}]}{\sum_{\mathbf{p}' \in \mathcal{B}} \overline{\mathrm{M}}_\theta[\mathbf{w}, \mathbf{p}']} \quad (17)$$

where $\mathrm{M}[\mathbf{p}, \mathbf{w}, \mathbf{W}_i, \mathbf{B}] = |\{j : \mathbf{b}_j = \mathbf{p} \text{ and } \mathbf{w}_j^i = \mathbf{w}\}|$ is the number of times word $\mathbf{w}$ appears in $\mathbf{W}_i$ aligned with the pronunciation $\mathbf{p}$. Conceptually, each path of pronunciations $\mathbf{B}$ contributes a soft-vote towards each of its constituent pronunciations according to its posterior probability. To ease the notation we have not explicitly required the number of pronunciations to be equal to the number of words, however, this is easy to enforce in practice. Once converged, θ^* can be used during decoding as the weights of our new stochastic lexicon.

In the continuous case above, the log-likelihood is not necessarily concave and we therefore cannot guarantee that the parameters reach a global optimum. The situation is different regarding isolated words. The isolated word case is also instructive with respect to understanding the model, and is thus presented here for completeness. Suppose we have a data set of example utterances of isolated words, $D_I = \{\mathbf{u}_i, \mathbf{w}_i\}$. The optimization of the log-likelihood in (14), would then reduce to the following:

$$\arg\max_\theta \mathcal{L}(\theta|D_I) = \arg\max_\theta \sum_{i=1}^{M} \log \sum_{\mathbf{b} \in \mathcal{B}} P(\mathbf{u}_i|\mathbf{b}, \mathbf{w}_i) \cdot \theta_{\mathbf{b}|\mathbf{w}_i} \quad (18)$$

Now that there is no dependency between words in the same utterance, the product of parameters is gone and it is simple to show that the log-likelihood in (18) along with the normalization constraints is concave. Amongst other techniques, EM can be used to optimize the log-likelihood. Furthermore, the update equations may be slightly easier to interpret than for the more general case.

$$\mathrm{E-step}: \quad P(\mathbf{p}|\mathbf{u}_i, \mathbf{w}_i; \theta) = \frac{p(\mathbf{u}_i|\mathbf{p}, \mathbf{w}_i) \cdot \theta_{\mathbf{p}|\mathbf{w}_i}}{\sum_{\mathbf{b}} p(\mathbf{u}_i|\mathbf{b}, \mathbf{w}_i) \cdot \theta_{\mathbf{b}|\mathbf{w}_i}} \quad (19)$$

$$\mathrm{M-step}: \quad \theta^*_{\mathbf{p}|\mathbf{w}} = \frac{1}{M_{\mathbf{w}}} \sum_{j:\mathbf{w}_j = \mathbf{w}} P(\mathbf{p}|\mathbf{u}_j, \mathbf{w}; \theta) \quad (20)$$

In the expectation step of the isolated word case, the generative nature of the pronunciation mixture model becomes clear. With the help of Bayes' rule, we compute the posterior probability that pronunciation $\mathbf{p}$ generates an acoustic signal $\mathbf{u}_i$ given that it contains the word $\mathbf{w}_i$. In the maximization step, the parameter representing the probability of a pronunciation $\mathbf{p}$ given $\mathbf{w}$ is found by normalizing the sum of the posteriors for that pronunciation for each word. A simple interpretation of these equations for a given word is that each utterance is allowed a *soft-vote* of $1/M_{\mathbf{w}}$ to distribute among a set of candidate pronunciations, where $M_{\mathbf{w}}$ is the number of utterances in the training data that contain the isolated word $\mathbf{w}$.

VII. Practical Application of the PMM

Even in the isolated word case, there are many aspects to implementing the pronunciation mixture model that we have so far left unspecified. What does one use to initialize θ? How can we train a graphone language model given a phoneme-based lexicon? Can we implement the equations of Section VI in a tractable way given that they suggest summing over an infinitude of possible pronunciations? What optimizations can one make for efficiency? This section provides insight into how we have answered these questions for the experiments described in the remainder of this paper, as well as how one might explore other variations of the PMM framework.

Many of the questions posed above are related, at least to some degree, to the initialization of the parameter values. Note that only the non-zero values can remain non-zero throughout the EM training procedure. The pronunciations corresponding to the non-zero values form the *support* of a stochastic lexicon. It is vital, therefore, to choose the support so that it both contains the pronunciations most likely to receive high weights during training *and* remains small enough that the computation is tractable.

One way to achieve these goals is to allow experts to specify a set of pronunciations, but leave the weighting of these pronunciations to the PMM. Conventional wisdom suggests that weighting pronunciations does not achieve significant gains. This may be due, in part, to the fact that hand-crafted lexicons have relatively few pronunciations per word. One way to alleviate this sparsity is to make use of phonological rules, such as those described in Section III. With such rules, it is straight-forward to expand phonemic baseforms provided by experts to a set of pronunciations at the phone-level. We can initially give each word's pronunciations uniform probabilities, and iteratively re-weight them using the PMM.

Ultimately, though, our hope is to apply the PMM technique to words unseen by expert eyes and unheard by expert ears. To do this, we elicit the help of a letter-to-sound system to provide the initial support for the PMM. In theory, a PMM could be applied to pronunciations generated from any L2S system, so long as the L2S system was able to hypothesize more than one pronunciation per word. In this work, we use the state-of-the-art joint-sequence models described in Section V. As described in Section II, however, a typical configuration of our recognizer employs a lexicon represented in terms of phonemes rather than phones, made possible by the use of manually crafted phonological rules that expand sequences of phonemes to all their possible phone realizations. Ideally, we would like to bypass these phonological rules altogether. The FST implementation of the recognizer would then be simplified to $R = C \circ L \circ G$ where L is now a lexicon FST that maps phone sequences to words. To accomplish this, however, we must first train a joint-sequence model at the phone level, i.e. a graphone model.

To generate graphone models certain manipulations must be made in order to coerce the baseline expert lexicon into a form suitable for the training of a phone-based L2S model. First, we employ direct expansion of the phoneme lexicon using phonological rules. This may not, however, adequately account for phonotactics across word boundaries in continuous speech. For example, the co-articulation that occurs at the boundary in the phrase *gas shortage* often results in the palatalization of the alveolar fricative that ends the word *gas*. Part of the purpose of phonological rules is to account for these types of cross-word phenomena. One can therefore *also* expand the transcripts of a training set with these same phonological rules, maintaining word boundary markers. It is then possible to accumulate the resulting phone-level pronunciations for inclusion in our training corpus. Pronunciations generated in this fashion will have phonological rules applied across word boundaries, and thus exhibit phones sequences consistent with their context in the continuous speech.

With a graphone language model in hand we can initialize the PMM by setting $\theta_{\mathbf{b}|\mathbf{w}}$ to be the graphone language model's $P(\mathbf{w}, \mathbf{b})$ of (9). It may, at first, be of some concern that this model is not in the conditional form θ is meant to represent. Fortunately, the normalization of $P(\mathbf{w}, \mathbf{b})$ that induces the appropriate conditional distribution merely requires a constant with respect to the maximization of θ and can therefore be ignored. This means that in practice, with a sufficiently small graphone language model, we can compute the posteriors needed during EM in the following manner. Let us first examine the isolated word case. For a particular word $\mathbf{w}$ we can create an FST $H_{\mathbf{w}}$ that represents the graphone language model restricted to the spelling of $\mathbf{w}$. The probabilities $p(\mathbf{u}_i|\mathbf{b}, \mathbf{w}_i)$ in (19) are then simply recovered through a recognition task in which the search space is constructed as $R_{\mathbf{w}} = C \circ J \circ H_{\mathbf{w}}$, where J is simply a mapping from phones to graphones. The recognition scores and the N-best list of pronunciations are then normalized to yield the posteriors. In subsequent iterations, the graphone language model is no longer used; instead, only the pronunciations represented in the N-best lists are considered.

For large graphone language models, directly manipulating FSTs can become unwieldy. This becomes especially apparent in the case of continuous speech, where word boundaries must also be considered. In this case, it is often simpler to generate the support for the lexicon directly by choosing the K most likely paths, and thus pronunciations, from the graphone language model. These N-best lists can be converted into FSTs as substitutes for $H_{\mathbf{w}}$, allowing for direct control over the size of the lexicon's support. In the continuous word case, the recognition search space, $R_{\mathbf{W}}$, can be constructed for a particular transcript $\mathbf{W}$ by concatenating each word's $H_{\mathbf{w}}$ pronunciations with word delimiters, and then performing the necessary compositions to ensure that context-dependent input labels are mapped to graphone output labels for the transcript $\mathbf{W}$. Normalizing the log-probabilities across the entries of the N-best list found during recognition yields the necessary posterior, $P(\mathbf{B}|\mathbf{u}_i, \mathbf{W}_i; \theta)$, of (16). Again, in subsequent iterations the dependency on the L2S model is dropped and only the pronunciations found in the N-best list are considered.

A number of approximations can be made for efficiency. While technically, recognition should be performed at each iteration to generate a new set of paths and posteriors for the expectation-step, in practice, we have found it more efficient to generate a large N-best list on the first iteration, and simply re-score the paths during subsequent iterations. This requires that the recognizer output the acoustic and language model scores separately, so that the acoustic scores can be multiplied with appropriate terms from the stochastic lexicon at each iteration. A final strategy, not explored in this work, is to perform coordinate ascent on the log likelihood. We might, for instance, fix the parameters of every word except for one, $\hat{\mathbf{w}}$. Utterances that contain $\hat{\mathbf{w}}$ would effectively be force-aligned with respect to the surrounding words, and the full pronunciation space of $\theta_{\mathbf{b}|\hat{\mathbf{w}}}$ would be explored using the PMM. This could continue for all words in the training set.

Once training is complete, it may be unrealistic to include every pronunciation with a non-zero $\theta_{\mathbf{b}|\mathbf{w}}$ in the final lexicon. There are a number of appealing ways one might filter pronunciations based on these learned weights. In our previous work, we chose to remove all pronunciations with weights below a certain threshold, with the caveat that all words must have at least one pronunciation. Here, we adopt the same approach, ensuring that all pronunciations $\theta_{\mathbf{b}|\mathbf{w}} \geq T$ are retained in the lexicon. The weights are then renormalized after thresholding.

VIII. Experimental Results

In our previous work, we experimented with both isolated word and continuous speech data. Since data in the latter form is

more readily available, we believe there is greater utility in exploring the continuous case more thoroughly. For this reason, the experiments in this work will expand upon those performed in [2] on the weather query corpus [30]. Whereas previously we fixed the L2S parameters used in the pronunciation mixture model, in this work we explore a variety of initializations. We further explore the flexibility of PMM initialization in an experiment that makes use of the expert lexicon to provide initial candidate pronunciations.

All of our experiments make use of the landmark-based speech recognizer described in Section III. The configuration used here takes MFCC averages over varying durations around the hypothesized landmarks to generate 112-dimensional feature vectors. These feature vectors are then whitened via a PCA rotation. The first 50 principal components are kept as the feature space over which diphone acoustic models are built. Each model is a diagonal Gaussian mixture with up to 75 mixture components trained on the weather query corpus training set, which consists of telephone speech.

The weather query corpus is composed of relatively short sentences, with an average of six words per utterance. After pruning the original training and test sets of all utterances containing non-speech artifacts, we end up with a 87,351 utterance training set amounting to 76.68 hours of speech. We divide the remaining utterances into a 3,497 utterance test set containing 3.18 hours of speech and a 1,179 utterance development set containing 0.84 hours of speech. The context-dependent acoustic models used with this corpus were trained using the expert lexicon in a speaker-independent fashion on a large data set of telephone speech of which this corpus is a subset.

The training set contains a vocabulary of 1,805 words, which comprise the lexical items used in constructing both the lexicon and language model. There are 71 words that occur exactly once in the training set, implying that a PMM only has one acoustic example of each of these words with which to generate a set of pronunciations. The most frequent word, *in*, on the other hand, occurs 33,967 times. For testing, a trigram language model is built using only the training set transcripts, and therefore contains exactly these 1,805 words. For efficiency reasons, however, a bigram trained in a similar fashion builds the initial lattice during decoding, and the trigram is then used in a subsequent re-scoring pass.

The expert-crafted lexicon used in these experiments is originally based on the publicly available PronLex dictionary [31]. The dictionary contains roughly 150,000 pronunciations specified at the *phoneme* level. As such, these pronunciations must be expanded with a set of phonological rules to be made compatible with the aforementioned diphone-based acoustic models. This can be performed during the composition of the recognizer's search space. Doing so, and decoding with this baseline setup on the test set yields the word error rate of 9.5% as reported in [2]. As described in the previous section, we can also expand the expert lexicon to a set of phone pronunciations and learn a set of weights using the PMM framework. While the initial expert lexicon had an average of 1.2 pronunciations per word (PPW), its expansion leads to a phone-based lexicon with a PPW of 4.0. Using these pronunciations in a PMM and pruning the result with $T = .01$ yields an "expert-PMM" with a PPW of 2.1. Decoding with this lexicon brings the WER down to 9.2%.

TABLE I
EVALUATION OF LEXICONS BUILT ON GRAPHONE LANGUAGE MODELS OF VARYING SIZES (M). FOR EACH LEXICON, WE USE A CUTOFF $T = .01$ TO THRESHOLD THE FINAL PRONUNCIATIONS INCLUDED DURING DECODING ON THE TEST SET. WE REPORT THE AVERAGE NUMBER OF PRONUNCIATIONS PER WORD (PPW) AS WELL AS RECOGNITION PERFORMANCE ON THE DEVELOPMENT AND TEST SETS. † IN THE $M = 1$ CASE, MANY WORDS DO NOT HAVE PRONUNCIATIONS WITH WEIGHTS $\geq T$, AND A LARGE NUMBER OF PRONUNCIATIONS ARE PRUNED AWAY

Singular Graphones and variable M					
	M=1	M=2	M=3	M=4	M=5
LM FST Size	28K	64K	624K	3.1M	9.4M
WER Using Graphones Alone (T=.01)					
PPW	3.9†	14.2	11.4	7.5	5.9
WER Dev.	71.4	20.5	14.3	13.1	12.5
WER Test	74.9	17.6	11.7	10.9	10.2
WER Using Graphone-based PMM (T=.01)					
PPW	6.08	4.0	3.5	3.1	3.0
WER Dev.	17.7	10.6	10.6	10.5	10.7
WER Test	15.6	8.4	8.2	8.1	8.2

In previous work, the joint-sequence models we explored did not vary with respect to their training parameters. We reported results regarding a 5-gram language model over both graphones and graphonemes, a configuration that had been shown to produce good results during recognition [23]. We showed that, for instance, using the graphone model to initialize the PMM yielded recognition results as good or better than its phoneme-based counterpart. Moreover the size of the final phone-based lexicon was significantly smaller, likely due to increased precision that can be employed during pruning based on pronunciation weights. In this work, therefore, we focus our attention on phone-based lexicons, effectively cutting out the need for phonemic baseforms and phonological rules during decoding. The lexicon used for graphone training in this work was expanded as described in the previous section and contains almost 425,000 phone pronunciations.

We perform the joint-multigram training algorithm described in [3] with a variety of parameter settings. Recall from Section V that we must choose the maximum number of graphemes, L, and phones, R, that appear in a graphone. We also must decide on the size of the M-gram we choose to train over the learned alignments. Following the lead of Bisani and Ney, we generate joint-sequence models with the L-to-1 and L-to-L graphone constraints, where $L = 1 \ldots 3$. We also vary language model size by training M-grams, where $M = 1 \ldots 5$. In previous work, we had restricted our attention to the $L = 1, M = 5$ case. Tables I and II delineate characteristics of trained models for other parameter settings. We show, for instance, the memory footprint of the FST representing graphone language models of each type. Naturally, the higher-order M-grams produce larger FSTs.

We experimented with values of N and K using our development set. Ideally, the length of the N-best list would contain as much of the probability mass represented in the lattice as possible. In our case, setting $N = 500$ was found to be more than sufficient. The number of initial pronunciations K is somewhat more difficult to experiment with, since it involves

TABLE II
EVALUATION OF LEXICONS BUILT ON JOINT-SEQUENCE MODELS WITH VARIOUS GRAPHONE SIZES (L-TO-R) AND A FIXED LANGUAGE MODEL SIZE OF $M = 5$. A THRESHOLD OF $T = .01$ IS USED TO PRUNE THE PRONUNCIATIONS USED FOR DECODING. WE REPORT THE AVERAGE NUMBER OF PRONUNCIATIONS PER WORD (PPW) ALONG WITH WER RATES ON THE DEVELOPMENT AND TEST SETS. ALSO SHOWN IN THIS TABLE ARE THE RELATIVE SIZES OF THE GRAPHONE LANGUAGE MODEL USED TO GENERATE PRONUNCIATIONS OR INITIALIZE THE PMM

Graphones with $M = 5$ and variable L-to-R					
	1-to-1	1-to-2	1-to-3	2-to-2	3-to-3
LM FST Size	9.4M	12M	12M	25M	21M
WER Using Graphones Alone (T=.01)					
PPW	5.9	5.7	5.7	5.4	5.5
WER Dev.	12.5	12.2	12.4	12.2	12.3
WER Test	10.2	10.2	10.3	10.4	10.1
WER Using Graphone-based PMM (T=.01)					
PPW	3.0	3.0	2.9	2.9	2.9
WER Dev.	10.7	10.6	10.5	10.4	10.6
WER Test	8.2	8.2	8.2	8.2	8.2

reconstructing the FST used for decoding during training. As described in the previous section, when taken to the limit K can actually vary for each word and be set to the number of pronunciations represented in a graphone language model. In our isolated word experiments [1], this was the approach we took. In this domain, we have found that initializing θ with the top $K = 200$ pronunciations from a graphone M-gram is more manageable from a computational perspective, and does not impact performance. For this work, we initialized the weights of our PMM with the normalized scores of the graphone language model.

Table I reports performance achieved on the weather corpus test set using lexicons based on singular graphone language models of varying size M. We begin by employing lexicons directly generated from joint-sequence models. To construct a lexicon for this task given a particular graphone language model FST, we over-generate an N-best list of pronunciations and their scores. We keep only those pronunciations that are consistent with our acoustic models, normalize their scores and then prune all pronunciations below $T = .01$. The remaining pronunciations, and their weights, are transformed into an FST and used during decoding. The general trend is consistent with the work of [14], which found that singular graphones tend to perform well in conjunction with higher order language models. It is also interesting to note how the average number of pronunciations per word varies with M. Here, the higher entropy of the lower order M-grams becomes apparent in the pronunciations that are generated. While $M = 1$ appears to be an exception, this is only due to our pronunciation pruning criteria of $T = .01$.

Table I also depicts the use of these singular graphone based lexicons to initialize a pronunciation mixture model. To initialize the PMM, we once again over-generate a set of pronunciations and their scores from the L2S model. These pronunciations are used to decode the training set, providing N-best lists necessary to perform the expectation maximization in the manner described in the previous section. Once EM has converged, the pronunciations and their scores are thresholded with $T = .01$ and compiled into an FST for decoding. We found that EM typically converged in three or four iterations on our development set, but decided to let it run out to the sixth iteration before compiling the lexicons for testing. Somewhat surprisingly, these results suggest that the higher-order M-grams become less critical when acoustic information is added into the equation. Indeed, we are able to achieve the best performance on this test set with $M = 4$.

TABLE III
A SUMMARY OF THE MAIN RESULTS ON THE WEATHER QUERY CORPUS

Summary on Test	
L2S Alone	10.1
Expert Lexicon	9.5
Expert PMM	9.2
L2S+PMM	8.2

We now turn to varying the size of the graphone units themselves. Once again, we break our analysis down into lexicons that are based solely on graphones, and those that incorporate acoustic information via the PMM. Table II provides PPW and WER statistics for lexicons built using various values of L and R while keeping M fixed at 5. Once again, the performance of graphone L2S models alone cannot compete with the analogous models that incorporate acoustic information through the PMM. Performance is roughly uniform across the different graphone sizes, suggesting that the convenience of working with the smaller, singular graphone language model is not detrimental to performance. These results are once again consistent with the work of [14]. Of course, it must be said that these experiments were performed on English, and that the parameters ideal for another language may be different.

Table III summarizes a few key results presented in this paper. In particular, we have found that while a hand-crafted dictionary can out-perform a state-of-the-art L2S model used in isolation, the story changes when acoustic information is incorporated into a lexicon's training procedure. As Table III shows, the L2S model supplemented with acoustic information outperforms the expert pronunciations, even when these pronunciations are trained in a similar fashion. Except for the small gain achieved by weighting the expert lexicon, the differences shown in this table are all found to be statistically significant using both Matched-Pair Sentence-Segment Word Error (MAPSSWE) and sentence-level McNemar tests [32], with $p < .005$.

IX. PHONETIC ANALYSIS

We now turn to analyzing the pronunciations generated in our stochastic lexicon. In this section, we use the pronunciations learned with the PMM initialized via a graphone language model with $L = 1$, $R = 1$, and $M = 5$. We first attempt to measure the degree to which the expert pronunciations are represented in the PMM. When expanded using the phonological rules, the 1,805 words in the weather corpus vocabulary were found to have a total of 7,324 pronunciations. Using the PMM to weight the expert lexicon and then pruning with $T = .01$ yields 3,863 pronunciations and slightly better performance. Interestingly, the graphone PMM with a similar threshold contains 5,495 phone-level pronunciations. Given its superior performance, this would suggest that the PMM is pruning superfluous, possibly incorrect pronunciations relative to the expert

TABLE IV
A PHONETIC COMPARISON BETWEEN THE EXPERT AND GRAPHONE PMMS

Word	Weighted Expert Pronunciations		PMM Pronunciations	
bangalore	0.38	bcl b aa ng g el ao r	0.53	bcl b ae ng g el ao r
			0.28	bcl b aa ng gcl g el ao r
general	0.40	dcl jh eh n axr el	0.54	dcl jh eh n axr el
general	0.30	dcl jh eh n r el	0.19	dcl jh eh n axr l
general	0.25	dcl jh eh n axr l	0.17	dcl jh eh n r el
istanbul	1.0	ih s tcl t aa n bcl b uw l	0.41	ih s tcl t ae n bcl b el
			0.20	ih s tcl t aa n bcl b el
			0.14	ih s tcl t ax n bcl b el
ottawa	0.95	aa dx ax w ax	0.51	aa dx ax w aa
	0.04	aa tcl t ax w ax	0.42	aa dx ax w ax
nice	0.75	n ay s	0.75	n ay s
	0.24	n iy s	0.24	n iy s

lexicon, but also is proposing new, beneficial pronunciations that do not appear in the expert PMM. These assertions are bolstered by the fact that of the 3,863 phone pronunciations found in the expert PMM, almost all of them (3,658) are still represented within the graphone PMM. On the other hand, of the 5,495 pronunciations in the graphone PMM, only 3,715 were found among the 7,324 original pronunciations in the expanded expert lexicon.

It is also interesting to note that the PMM-approach settled on pronunciations with fewer phones on average than were in the original expert lexicon. These shorter pronunciations are consistent with the fact that, while WER improves for the PMM, the number of insertions (460) actually increases over the number of insertions that occur when decoding with the PMM-weighted expert lexicon (192). Similar to our previous work [1], we found that phone substitutions often involved vowels, many of which were shortened to a schwa. These are, perhaps, indications that the PMM learns pronunciations compatible with continuous speech.

A more detailed analysis reveals that the expert and graphone PMMs are similar in many respects. In fact, 91.5% of the highest weighted pronunciations were the same in both lexicons. Table IV lists a few examples from each lexicon to supplement this quantitative comparison. Some of the differences found were vowel substitutions such as those seen in the word *bangalore*. Quite often, however, we found the graphone PMM shifting weights to cover pronunciation variations, such as in the case of the word *general* and *istanbul*. In the latter example, there is arguably a lexicon correction as well, where the expert lexicon posited /uw/, while the PMM much preferred /el/. With respect to the weights, it is interesting to note, that the graphone PMM contains examples that both sharpen the distribution over pronunciations, such as *general*, and spread out the probability mass to reasonable pronunciations not found in the expanded expert lexicon, as in *ottawa*. Finally, in both cases, the PMM can learn weights over heteronyms. This weather corpus contains sentences such as "*Where is a nice sunny place in the caribbean?*" and also "*Is it rainy in Nice, France today?*" We found that the learned pronunciations for the word *nice* correlated well with the frequency of each meaning of the word in context.

X. SUMMARY AND FUTURE WORK

This work has introduced a maximum likelihood, generative approach to incorporating acoustic examples in the pronunciation generation process. We have shown that a pronunciation mixture model can be used to weight an existing set of handcrafted pronunciations, and perhaps more importantly, may be used to reliably generate better-than-expert pronunciations for continuous speech. We believe that wrapping the lexicon into a statistical framework is a constructive step that presents exciting new avenues of exploration. We have also shown that the PMM can be trained on the same data as the acoustic and language models, and hence requires no additional resources. These properties make pronunciation training a cheap and effective additional tool for building an ASR system.

In previous work, we have shown that these findings extend to corpora with a variety of characteristics including those with noisy acoustics. We have demonstrated, for example, that crowd-sourcing platforms such as Amazon Mechanical Turk can be employed to collect pronunciations which, although noisy, can be used to generate pronunciations that improve recognition performance on clean speech. Such platforms are sometimes outfitted with APIs that allow developers to incorporate automated crowdsourcing into their algorithms. We have used one such API to enable a spoken language system to collect speech and improve pronunciations on-the-fly with the PMM [33].

We believe the possibility of learning better-than-expert baseforms in arbitrary domains opens up many research possibilities. There are two clear directions with respect to our training procedure that warrant further exploration in the immediate future. The first is to examine acoustic model and lexicon co-training in an iterative fashion, effectively taking a maximum-likelihood step along a set of coordinates in the probability space represented by the recognizer. In this context, it is not clear whether over-fitting would overshadow the potential advantages of such an approach. A second avenue is to move beyond maximum-likelihood, and explore discriminative approaches to pronunciation learning.

Finally, it is important to note that our current experiments still rely on the expert lexicon in order to train the L2S system. Our ongoing work aims to remove this dependency. An initial experiment that we have performed along these lines is to use a simple unweighted phone loop atop our diphone acoustic model to decode the training set. The resulting phone sequences can then be associated with grapheme sequences found in the corresponding transcripts. In precisely the way we train a graphone language model from a lexicon, we are then able to learn a graphone language model from a pseudo-lexicon built from these transcripts and phonetic decodings. The learned L2S can then be used to initialize a PMM. On the weather query corpus, the lexicon resulting from this technique performs surprisingly well on our test set, with a WER of 8.7%. While this preliminary experiment still makes use of acoustic models trained using the expert lexicon, we view this as a positive step towards the possibility making use of unsupervised acoustic units such as those in the work of Lee and Glass [34].

If it were feasible to simultaneously train the lexicon and discover an acoustic model, large vocabulary speech recognizers could be built for many different languages with little to no expert input. While this research may question the orthodox view that pronunciations need to be interpretable linguistically, our

hope is that it may be a positive step towards breaking the language barrier of modern speech recognition.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] I. Badr, I. McGraw, and J. R. Glass, "Learning new word pronunciations from spoken examples," in *Proc. INTERSPEECH*, 2010, pp. 2294–2297.
[2] I. Badr, I. McGraw, and J. R. Glass, "Pronunciation learning from continuous speech," in *Proc. INTERSPEECH*, 2011, pp. 549–552.
[3] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, May 2008.
[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[5] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 2, pp. 179–190, Mar. 1983.
[6] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Comput. Speech Lang.*, vol. 17, no. 2–3, pp. 137–152, 2003.
[7] I. L. Hetherington, "The MIT finite-state transducer toolkit for speech and language processing," in *Proc. INTERSPEECH*, 2004, pp. 2609–2612.
[8] I. L. Hetherington, "An efficient implementation of phonological rules using finite-state transducers," in *Proc. EuroSpeech*, 2001, pp. 1599–1602.
[9] T. J. Hazen, I. L. Hetherington, H. Shu, and K. Livescu, "Pronunciation modeling using a finite-state transducer representation," *Speech Commun.*, vol. 46, no. 2, pp. 189–203, 2005.
[10] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," *Speech Commun.*, vol. 29, no. 2–4, pp. 225–246, 1999.
[11] I. L. Hetherington, "A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding," Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, MA, 1995.
[12] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 288–298, Mar. 2001.
[13] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Proc. INTERSPEECH*, 2005, pp. 725–728.
[14] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, 2003, pp. 2033–2036.
[15] R. M. Kaplan and M. Kay, "Regular models of phonological rule systems," *Comput. Linguist.*, vol. 20, pp. 331–378, 1994.
[16] S. Seneff, "Reversible sound-to-letter/letter-to-sound modeling based on syllable structure," in *Proc. Human Lang. Technol.: Conf. North Amer. Chap. Assoc. for Comput. Linguist. (HLT-NACCL)*, 2007, pp. 153–156.
[17] Y. Marchand and R. I. Damper, "A multi-strategy approach to improving pronunciation by analogy," *Comput. Linguist.*, vol. 26, pp. 195–219, 2000.
[18] J. Bellegarda, "Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy," *Speech Commun.*, vol. 46, no. 2, pp. 140–152, 2005.
[19] A. Laurent, P. Delglise, and S. Meignier, "Grapheme to phoneme conversion using an SMT system," in *Proc. INTERSPEECH*, 2009, pp. 708–711.
[20] S. Jiampojamarn and G. Kondrak, "Online discriminative training for grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, 2009, pp. 1303–1306.
[21] V. Pagel, K. Lenzo, and A. W. Black, "Letter to sound rules for accented lexicon compression," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, 1998.
[22] J. Häkkinen, J. Suontausta, S. Riis, and K. J. Jensen, "Assessing text-to-phoneme mapping strategies in speaker independent isolated word recognition," *Speech Commun.*, vol. 41, no. 2–3, pp. 455–467, 2003.
[23] S. Wang, "Using graphone models in automatic speech recognition," M.S. thesis, Massachusetts Inst. of Technol., Cambridge, MA, 2009.
[24] D. McAllaster, L. Gillick, F. Scattone, and M. Newman, "Fabricating conversational speech data with acoustic models: A program to examine model-data mismatch," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, 1998.
[25] B. Maison, "Automatic baseform generation from acoustic data," in *Proc. INTERSPEECH*, 2003, pp. 2545–2548.
[26] G. F. Choueiter, M. I. Ohannessian, S. Seneff, and J. R. Glass, "A turbo-style algorithm for lexical baseforms estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 4313–4316.
[27] L. R. Bahl, S. Das, P. V. Desouza, M. Epstein, R. L. Mercer, B. Merialdo, D. Nahamoo, M. A. Picheny, and J. Powell, "Automatic phonetic baseform determination," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1991, pp. 173–176.
[28] X. Li, A. Gunawardana, and A. Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proc. Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2007, pp. 130–135.
[29] O. Vinyals, L. Deng, D. Yu, and A. Acero, "Discriminative pronunciation learning using phonetic decoder and minimum-classification-error criterion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2009, pp. 4445–4448.
[30] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington, "Jupiter: A telephone-based conversational interface for weather information," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 1, pp. 85–96, Jan. 2000.
[31] P. Kingsbury, S. Strassel, and R. MacIntyre, CALLHOME American English lexicon (PRONLEX), 1997, lDC Catalog No. LDC97L20.
[32] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1989, pp. 532–535.
[33] I. McGraw, S. Cyphers, P. Pasupat, J. Liu, and J. Glass, "Automating crowd-supervised learning for spoken language systems," in *Proc. INTERSPEECH*, 2012.
[34] C. Lee and J. R. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2012, pp. 40–49.

Ian McGraw received his B.S. in Computer Science from Stanford University in 2005, where his thesis work centered around improving inference algorithms on graphical models. He then received an S.M. from MIT in 2008 as a member of the Spoken Language Systems (SLS) group, where his research involved the use of speech recognition in computer aided language learning systems. His Ph.D. work, completed with SLS in 2012, employed crowd-sourcing techniques to automatically adapt spoken language systems.

Ibrahim Badr received his B.E. in Computer and Communications Engineering from the American University of Beirut in 2009. He then enrolled at the Massachusetts Institute Technology as a graduate student and member of the Spoken Language Systems (SLS) group. He was awarded an M.S. in 2011. His research centers on automatic pronunciation learning for speech recognition.

James R. Glass (M'78–SM'06) is a Senior Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) where he heads the Spoken Language Systems Group. He is also a Lecturer in the Harvard-MIT Division of Health Sciences and Technology. He received his B.Eng. in Electrical Engineering at Carleton University in Ottawa in 1982, and his S.M. and Ph.D. degrees in Electrical Engineering and Computer Science at MIT in 1985, and 1988, respectively. After starting in the Speech Communication group at the MIT Research Laboratory of Electronics, he has worked since 1989 at the Laboratory for Computer Science, and since 2003 at CSAIL. His primary research interests are in the area of speech communication and human-computer interaction, centered on automatic speech recognition and spoken language understanding. He has lectured, taught courses, supervised students, and published extensively in these areas. He is currently a Senior Member of the IEEE, and an associate editor for IEEE TRANS. AUDIO, SPEECH, AND LANGUAGE PROCESSING and for Computer, Speech, and Language.