# Pronunciation Learning from Continuous Speech

Ibrahim Badr, Ian McGraw, and James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA

{iab02, imcgraw, glass}@mit.edu

## Abstract

This paper explores the use of continuous speech data to learn stochastic lexicons. Building on previous work in which we augmented graphones with acoustic examples of isolated words, we extend our pronunciation mixture model framework to two domains containing spontaneous speech: a weather information retrieval spoken dialogue system and the academic lectures domain. We find that our learned lexicons out-perform expert, hand-crafted lexicons in each domain.

**Index Terms**: grapheme-to-phoneme conversion, pronunciation models, lexical representation

## 1. Introduction

The lexicon remains a peculiarity in the modern speech recognizer in that, unlike the language and acoustic models, data does not typically underpin its creation. Instead, the lexicon is usually a long list of hand-crafted rules in the form of dictionary entries that map a word to one or more *canonical* pronunciations. A data-driven approach to lexicon generation might discard the notion of canonicalization altogether, and instead generate a stochastic lexicon with pronunciations weighted according to learned statistics. Like the acoustic and language models, ideally pronunciations would be learned from data closely matching the test domain. In this work, we explore learning pronunciations from continuous speech data using a pronunciation mixture model (PMM).

The literature is replete with investigations of various letter-to-sound models to predict pronunciations of new words [1, 2, 3]. These approaches, however, are limited by the underlying expert lexicons upon which they are trained. Our work follows the examples of [4, 5, 6, 7] which use spoken examples to refine pronunciations. The work of [6], for example, adapts a graphoneme model's weights using acoustic examples and is applied to a name recognition task, while the work of [7] uses forced alignment 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. Curiously, this type of work is rarely applied across the entire lexicon to regularize the pronunciations with respect to the underlying acoustic models.

By contrast, in our work we learn pronunciations across all lexical entries rather than the few out-of-vocabulary words for which we do not have an expert opinion. Thus, our test experiments directly compare a learned stochastic lexicon with manually-crafted pronunciations. Although expert-lexicons typically outperform state-of-the-art letter-to-sound models, we show that incorporating spoken examples can yield pronunciations that are as good, if not better, than their hand-crafted counterparts. We use a generative approach similar to [6] to train the lexicon. Unlike previous work, however, we train pronunciations for all words on the same continuous speech used to learn the acoustic models. Despite the fact that the acoustic models were trained using the expert pronunciation dictionary, we are still able to show a significant improvement over experts with a learned lexicon. We see this work as a step towards being able to train a speech recognizer entirely from an orthographically transcribed corpus.

## 2. Pronunciation Modeling

Pronunciation variation has been identified as a major cause of errors for a variety of ASR tasks. Pronunciations are typically modeled in a speech recognizer by phonemic dictionary which may be accompanied by a set of rewrite rules to account for phonological variation.

The ASR system used in this paper incorporates manually crafted phonological rules that account for segmental mismatches between the underlying phonemic baseforms and surface-level phonetic units. These rules have been shown to outperform relying on context-dependent acoustic models to implicitly model phonetic variation [8].

In this work, we model the ASR's search space using a weighted finite-state transducer (FST) [9]. The 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:

$$N = C \circ P \circ L \circ G \tag{1}$$

In this paper, we first experiment with learning context-independent phoneme pronunciations along with their weights. That is, we try to replace the manually crafted FST $L$ while keeping the pronunciations rules FST $P$ unchanged. In a second experiment, we explore learning phone pronunciations directly, thus avoiding the use of the phonological rewrite rules altogether.

## 3. Grapheme-to-Phoneme Conversion

We utilize the joint-multigram approach employed in [1, 3] to model the relationship between graphemes and phonetic units. In this work, we use the term *graphone* to denote a model that maps graphemes to phones and *graphoneme* to refer to a model that maps graphemes to phonemes.

We begin by constructing an $n$-gram model over graphoneme sequences. We let $\mathbf{w}$ denote a grapheme sequence drawn from the set of all possible grapheme sequences $\mathcal{W}$ and $\mathbf{b}$ denote a phoneme sequence drawn from the set of all possible phoneme sequences, $\mathcal{B}$. A joint model of the letter-to-sound task can be formalized as:

$$\mathbf{b}^* = \arg\max_{\mathbf{b} \in \mathcal{B}} P(\mathbf{w}, \mathbf{b}) \tag{2}$$

Generally speaking, a graphoneme, $g = (w, b) \in \mathcal{G} \subseteq \mathcal{W} \times \mathcal{B}$, is a sub-word unit that maps a grapheme subsequence, $w$, to a phoneme subsequence, $b$. By analogy, a graphone is an alternative sub-word unit that maps a grapheme subsequence to a *phone* subsequence. In this work, we restrict our attention to singular graphones or graphonemes, in which a mapping is made between at most one grapheme and at most one phonetic unit. The empty subsequence $\epsilon$ is allowed, however a mapping from $\epsilon$ to $\epsilon$ is omitted. Taken together, a sequence of graphonemes, $\mathbf{g}$, inherently specifies a unique sequence of graphemes $\mathbf{w}$ and phonemes $\mathbf{b}$; however, there may be multiple ways to align the pair $(\mathbf{w}, \mathbf{b})$ into various graphoneme sequences $\mathbf{g} \in S(\mathbf{w}, \mathbf{b})$. The following table shows two possible graphoneme segmentations of the word "couple".

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\mathbf{w}$ | = | c | o | u | p | | l | e |
| $\mathbf{b}$ | = | k | ah | | p | ax | l | |
| | = | k | | ah | p | ax | l | |
| $\mathbf{g}_1$ | = | c/k | o/ah | u/$\epsilon$ | p/p | $\epsilon$/ax | l/l | e/$\epsilon$ |
| $\mathbf{g}_2$ | = | c/k | o/$\epsilon$ | u/ah | p/p | $\epsilon$/ax | l/l | e/$\epsilon$ |

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].

$$P(w, b) = \sum_{g \in S(w,b)} P(g) \approx \max_{g \in S(w,b)} P(g) \tag{3}$$

In our work, we use the open source implementation provided by [3], which runs the Expectation-Maximization (EM) algorithm on a training corpus of word-phoneme pronunciation pairs to automatically infer graphoneme alignments. We then train a standard 5-gram language model over the automatically segmented corpus of graphonemes. This configuration has been shown to produce good results for singular graphonemes [10].

# 4. Pronunciation Mixture Model

In [11], we use the joint distribution learned from the lexicon as described above to seed a pronunciation mixture model (PMM), which employs EM to iteratively update a set of parameters based on a example utterances. With the PMM approach, we learn a distribution over pronunciations for a particular word by treating them as components in a mixture and aggregating the posteriors across spoken examples. Beginning with the parametrization $\theta_{\mathbf{w},\mathbf{b}} = P(\mathbf{w}, \mathbf{b})$ in the isolated word case, we can characterize the log likelihood of a data set $(u_1^M, \mathbf{w})$ as follows:

$$L(\theta) = \sum_{i=1}^{M} \log\ p(u_i, \mathbf{w}; \theta) = \sum_{i=1}^{M} \log \sum_{\mathbf{b} \in \mathcal{B}} \theta_{\mathbf{w},\mathbf{b}} \cdot p(u_i | \mathbf{w}, \mathbf{b})$$

For each word $\mathbf{w}$ the EM algorithm can be run to iteratively update pronunciation weights, which may then be used in a stochastic lexicon. The following equations specify the expectation and maximization steps respectively:

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

M-step:
$$\theta^*_{\mathbf{w},\mathbf{b}} = \frac{1}{M} \sum_{i=1}^{M} P(\mathbf{b}|u_i, \mathbf{w}; \theta)$$

# 5. PMM for Continuous Speech

Extending this approach to continuous speech requires additional considerations. We once again model the sequence of phonetic units as a hidden variable, however, the general ASR problem is now a search for the most likely string of words $\mathbf{W}^* = \mathbf{w}_1^*, \cdots, \mathbf{w}_k^*$ given an utterance $u$:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}|u) = \arg\max_{\mathbf{W}} \sum_{\mathbf{B} \in \mathcal{B}^*} P(\mathbf{W}, \mathbf{B}|u) \tag{4}$$

where now, $\mathbf{B} \in \mathcal{B}$ is a sequence of phonemes or phones that might span multiple words and include silence. Thus, for the continuous case, we consider silence to be a phonetic unit and denote it with a '-'. For example, a possible phoneme sequence $\mathbf{B}$ for an utterance with transcription "the boy ate the apple" could be "dh ax b oy - - ey td - dh ah ae p ax l".

Equation 4 can be decomposed as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_{\mathbf{B} \in \mathcal{B}} P(\mathbf{W}) P(\mathbf{B}|\mathbf{W}) P(u|\mathbf{W}, \mathbf{B}) \tag{5}$$

where $P(\mathbf{W})$ is the language model, $P(\mathbf{B}|\mathbf{W})$ can be computed using a stochastic lexicon and $P(u|\mathbf{W}, \mathbf{B})$ can be approximated with the acoustic model $P(u|\mathbf{B})$. Note that the speech recognizer used in this paper uses standard Viterbi approximations during decoding. This reduces Equation 5 to the following:

$$\mathbf{W}^* = \arg\max_{\mathbf{W},\mathbf{B}} P(\mathbf{W}) P(\mathbf{B}|\mathbf{W}) P(u|\mathbf{B}) \tag{6}$$

### 5.1. EM for Estimating Phoneme Pronunciations

We now extend the Pronunciation Mixture Model (PMM) framework developed for isolated word recognition [11] to learn the appropriate weights that can model $P(\mathbf{B}|\mathbf{W})$ in continuous speech.

Our training data is comprised of $M$ utterances and their transcriptions $\{u_i, \mathbf{W}_i\}$ where $\mathbf{W}_i = w_1^i, \cdots, w_{k_i}^i$. We parametrize the log-likelihood as follows:

$$\sum_{i=1}^{M} \log P(u_i, \mathbf{W}_i|\theta) = \sum_{i=1}^{M} \log \sum_{\mathbf{B} \in \mathcal{B}} \sum_{\psi \in \Psi(\mathbf{W}_i, \mathbf{B})} P(u_i, \mathbf{W}_i, \mathbf{B}, \psi|\theta)$$

where $\psi$ is an additional hidden variable defined to segment the phonetic sequence $\mathbf{B}$ into $k$ baseforms while deleting the silences. Thus, $\psi$ is drawn from possible segmentations $\Psi(\mathbf{W}_i, \mathbf{B})$ and can be indexed to retrieve a particular word-baseform pair. For example:

- $\psi_1$(dh ax b oy - - ey td - dh ah ae p ax l) = dh ax
- $\psi_2$(dh ax b oy - - ey td - dh ah ae p ax l) = b oy
- $\psi_3$(dh ax b oy - - ey td - dh ah ae p ax l) = ey td
- $\psi_4$(dh ax b oy - - ey td - dh ah ae p ax l) = dh ah
- $\psi_5$(dh ax b oy - - ey td - dh ah ae p ax l) = ae p ax l

We can now further decompose the term as follows:

$$P(u_i, \mathbf{W}_i, \mathbf{B}, \psi) = P(u_i|\mathbf{B}) P(\mathbf{w}_1^i, \cdots, \mathbf{w}_{k_i}^i, \mathbf{b}_1, \cdots, \mathbf{b}_{k_i})$$

where $\mathbf{b}_i = \psi_i(\mathbf{B})$. Our acoustic models are trained such that when $\mathcal{B}$ is the *phoneme* alphabet, a pronunciation $\mathbf{b}_i$ is context independent and the equation above can be rewritten as:

$$P(u_i, \mathbf{W}_i, \mathbf{B}, \psi) = P(u_i|\mathbf{B}) \prod_{j=1}^{k_i} \theta_{\mathbf{w}_j^i, \psi_j(\mathbf{B})} \tag{7}$$

where $\theta_{\mathbf{w}_j^i,\mathbf{b}_j} = P(\mathbf{w}_j^i, \mathbf{b}_j)$. Our log-likelihood then becomes:

$$\sum_{i=1}^{M} \log P(u_i, \mathbf{W}_i|\theta) = \sum_{i=1}^{M} \log \sum_{\mathbf{B}\in\mathcal{B}} \sum_{\psi\in\Psi(\mathbf{W}_i,\mathbf{B})} P(u_i|\mathbf{B}) \prod_{j=1}^{k_i} \theta_{\mathbf{w}_j^i,\psi_j(\mathbf{B})}$$

The parameters, $\theta$, are initialized to our graphoneme n-gram model scores and multiple iterations of the EM algorithm are run.

E-step:

$$\overline{\mathrm{M}}_\theta[\mathbf{w},\mathbf{p}] = \sum_{i=1}^{M} \sum_{\mathbf{B}\in\mathcal{B}} \sum_{\psi\in\Psi(\mathbf{W}_i,\mathbf{B})} P(\mathbf{B},\psi|u_i,\mathbf{W}_i,\theta)\mathrm{M}[\mathbf{p},\mathbf{w},\mathbf{W}_i,\mathbf{B},\psi]$$

M-step:

$$\theta^*_{\mathbf{w},\mathbf{p}} = \frac{\overline{\mathrm{M}}_\theta[\mathbf{w},\mathbf{p}]}{\sum_{\mathbf{w}',\mathbf{p}'\in V\times\mathcal{B}} \overline{\mathrm{M}}_\theta[\mathbf{w}',\mathbf{p}']}$$

where $\mathrm{M}[\mathbf{p},\mathbf{w},\mathbf{W}_i,\mathbf{B},\psi]$ is the number of times word $\mathbf{w}$ appears in $\mathbf{W}_i$ aligned with the pronunciation $\mathbf{p}$.

The weights learned are directly used in a stochastic lexicon for decoding continuous speech. The term $P(\mathbf{B}|\mathbf{W})$ in Equation 5 can be computed as:

$$P(\mathbf{B}|\mathbf{W}) = \sum_{\psi\in\Psi(\mathbf{W},\mathbf{B})} \prod_{j=1}^{k} \frac{\theta_{\mathbf{w}_j,\psi_j(\mathbf{B})}}{\sum_{\mathbf{p}\in\mathcal{B}} \theta_{\mathbf{w}_j,\mathbf{p}}} \tag{8}$$

### 5.2. EM for Estimating Phone Pronunciations

The phonetic rules our recognizer applies between the phoneme-based lexicon and the context-dependent acoustic models create context dependencies across words at the *phone* level. A misguided attempt to apply the PMM directly to learning phone pronunciations would ignore the independence assumption made in Equation 7, which is no longer valid. In this section, we explore a model that assigns a probability to a word-pronunciation pair given the last phoneme of the previous word's pronunciation:

$$\begin{aligned} P(\mathbf{w}_1,\mathbf{b}_1,\ldots,\mathbf{w}_n,\mathbf{b}_n) &= \prod_{i=1}^{n} P(\mathbf{w}_i,\mathbf{b}_i|\mathbf{b}_{i-1}) \\ &= \prod_{i=1}^{n} P(\mathbf{w}_i,\mathbf{b}_i|LAST(\mathbf{b}_{i-1})) \end{aligned}$$

Where $LAST(\mathbf{b})$ is the last phone in the phone sequence $\mathbf{b}$. Here we make several independence assumptions, the first is similar to the Markov assumption made in n-gram language modeling, the second references the fact that only the ending of the previous word's pronunciation can affect the current word's pronunciation. Our new features $\theta_{\mathbf{w},\mathbf{b}|p} = P(\mathbf{w},\mathbf{b}|p)$, where $\mathbf{b} \in \mathcal{B}$ and $p$ is a single phone, can now be used in equation 7 as follows:

$$P(u_i,\mathbf{W}_i,\mathbf{B},\psi) = P(u_i|\mathbf{B}) \prod_{j=1}^{|W_i|} \theta_{\psi_j(\mathbf{B}),\mathbf{w}_j^i|LAST(\psi_{j-1}(\mathbf{b}))} \tag{9}$$

## 6. Experimental Setup

Experiments using the SUMMIT landmark-based speech recognizer [12] were conducted in two domains: a weather query corpus [13] and an academic lecture corpus [14].

### 6.1. Experimental Procedure

To evaluate the performance of our PMM model we used the following procedure for both the weather and lecture domains. We begin by cleaning the acoustic model training set by removing utterances with non-speech artifacts to generate a training set for PMM pronunciations. We then prepare two recognizers, the first based on manually created pronunciations of all the words in the training set and the second a learned PMM recognizer that contains all the pronunciations generated by our PMM model. We then compare the Word Error Rate (WER) of both recognizers on a common test set. Thus, both recognizers use precisely the same vocabulary, but the pronunciations are chosen or weighted either by human or machine. Although the expert-lexicon leaves pronunciations unweighted, it does include a number of entries with multiple pronunciations. To keep the number of PMM pronunciations included in the search space to a reasonable size we use a 0.005 threshold to prune out low probability pronunciations.

The weather query corpus is comprised of relatively short utterances, with an average of 6 words per utterance. After pruning the original training and test sets of all utterances containing non-speech artifacts, we ended up with a 87,600 utterance training set with an 1,805 word vocabulary and a 3,497 utterance test set. The acoustic models used with this corpus were trained on a large data set of telephone speech of which this corpus is a subset.

The lecture corpus contains audio recordings and manual transcriptions for approximately 300 hours of MIT lectures from eight different courses and nearly 100 MITWorld seminars given on a variety of topics [14]. The lecture corpus is a difficult data set for ASR systems because it contains many disfluencies, poorly organized or ungrammatical sentences, and lecture specific key words [15]. Compared to the weather corpus the sentences are much longer, with about 20 words per utterance on average.

As in the weather domain, we discard utterances that contain non-speech artifacts from the training set used to train the acoustic model end up with a 50K utterance training set. We then cleaned the remaining utterances to create a 6,822 utterance test set. We report results on training pronunciations and decoding with back-off maximum likelihood trained Acoustic Models [15]. We leave the use of discriminatively trained models for future work. The back-off maximum likelihood model uses a set of broad phonetic classes to divide the classification problem originating from context-dependent modeling into a set of subproblems. The reported results differ from those in paper [15] because we use a 25K word vocabulary of all the words in our training set. The original paper uses a 35K word vocabulary with some words absent from the training set.

### 6.2. Graphone/Graphoneme Training

A 150,000 word dictionary of manually generated phoneme pronunciations was used to train the graphoneme $n$-gram parameters according to the procedures described in [10].

To train our graphone model, one might be tempted to simply expand the phoneme-based lexicon according to the pronunciation rules learned in Section 2. Unfortunately, given the manner in which our acoustic models were trained, the beginnings and endings of pronunciations are context-dependent at the phone level. Thus, we must expand all the sentences in our weather and lecture training corpora first to their phoneme pronunciations using the manually crafted dictionary and then to their phone variations using the pronunciation rules. These

| | Weather | Lectures |
|---|---|---|
| Graphoneme L2S | 11.2 | 47.6 |
| Expert | 9.5 | 36.7 |
| Phoneme PMM | 8.3 | 36.3 |
| Phone PMM | 8.3 | 36.1 |
| Context PMM | 8.3 | 37.7 |

Table 1: Results in WER (%)

phone pronunciations were properly generated since the pronunciation rules had access to the context of a word in a sentence.

## 7. Experimental Results

The results for learning pronunciations can be seen in Table 1. A baseline using a lexicon generated directly from graphonemes is shown to be significantly worse than both experts and the PMM. More interestingly, phoneme PMM pronunciations achieve more than a 1.2% WER reduction on the weather test set and a 0.4% WER reductions on the lectures test set over the hand-crafted lexicon. Both results were deemed statistically significant ($p<0.001$) using the Matched Pair Sentence Segment Word Error test. These results were achieved by running the EM algorithm until convergence which took around 8 iterations. It is important to note that we are learning these pronunciations without human supervision and hence this technique can be reliably used to predict pronunciations for out-of-vocabulary words from continuous speech. We also show improvement over a baseline of expert crafted pronunciations by training on the same data used for training our acoustic models. This suggests that not only are we learning better-than-expert pronunciations, we are also allowing more information to be extracted from the training set that can complement the acoustic models. This smaller size vocabulary of the weather domain could explain the higher gains achieved: since the PMM includes multiple pronunciations per word, this might make them more confusable a fact that is more apparent in the 25k vocabulary Lecture domain.

We also test on learning phone pronunciations in a similar context-independent setup as that of phonemes (Section 5.1). The results are referenced as "Phone PMM" in Table 1. One advantage of learning phone pronunciations in a context-independent setup is that we are no longer using the pronunciation rules that might be over-expanding the search space. This fact is made apparent in Table 2 where we see an increase in the number of pronunciations per word but a smaller search space.

A second reason to favor removal of phonological rules from the recognizer is simply that, when the lexicon is trained appropriately, they appear to be an unnecessary complication. It is also interesting to note that Phone PMM training is faster and requires only 4 EM iterations to achieve convergence. The disadvantage of the direct expansion approach we have described so far is that phone pronunciations are context-dependent with respect to the underlying acoustic models, a fact that is not represented in the learned lexicon.

We tried to model these cross-word dependencies by using the context-dependent model described in section *Context PMM*. Whereas the move from graphonemes to graphones reduces the complexity of our recognizer, incorporating context dependent models grows the search space, as seen in Table 2. From the results in table Table 1, however, it is clear that they are an unnecessary complication, and even hurt performance in the lectures domain. The degradation is likely due to a problem of context sparsity, which is caused by a greater mismatch between training and test in the lecture corpus.

| Weather Lexicon | Avg # Prons | # States | # Arcs | Size |
|---|---|---|---|---|
| Expert | 1.2 | 32K | 152K | 3.5 MB |
| Phoneme PMM | 3.15 | 51K | 350K | 7.5 MB |
| Phone PMM | 4.0 | 47K | 231K | 5.2 MB |
| Context PMM | 63 | 253K | 936K | 22 MB |
| **Lecture Lexicon** | **Avg. # Prons** | **# States** | **# Arcs** | **Size** |
| Expert | 1.2 | 226K | 1.6M | 36 MB |
| Phoneme PMM | 1.8 | 501K | 7.6M | 154 MB |
| Phone PMM | 2.3 | 243K | 1.2M | 28 MB |
| Context PMM | 8.9 | 565K | 2.7M | 61MB |

Table 2: Lexicon statistics for the weather and lecture domains.

## 8. Summary and Future Work

In this work we have shown that training a lexicon from continuous speech can yield improvements in WER when compared to a recognizer using a hand-crafted alternative. There are two clear directions in which we hope to further this work. The first is to co-train the acoustic models and the lexicon in an iterative fashion, effectively taking a maximum-likelihood step along a set of coordinates in the probability space represented by the recognizer. The second is to move beyond maximum-likelihood, and explore discriminative approaches to pronunciation learning on continuous speech. Regardless, we believe that wrapping the lexicon into a statistical framework is a constructive step, which presents exciting new avenues of exploration.

## 9. Acknowledgements

This work is sponsored by Nokia and the T-party project, a joint research program between Quanta Computer Inc. and MIT.

## 10. References

[1] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. Eurospeech*, 2003.

[2] S. Seneff, "Reversible sound-to-letter/letter-to-sound modeling based on syllable structure," in *Proc. HLT-NAACL*, 2007.

[3] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[4] L. R. Bahl and et al., "Automatic phonetic baseform determination," in *Proc. ICASSP*, 1991.

[5] B. Maison, "Automatic baseform generation from acoustic data," in *Proc. Eurospeech*, 2003.

[6] X. Li, A. Guanawardana, and A. Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proc. ASRU*, 2007.

[7] O. Vinyals, L. Deng, D. Yu, and A. Acero, "Discriminative pronunciation learning using phoentic decoder and minimim-classification-error criterion," in *Proc. ICASSP*, 2009.

[8] T. Hazen, L. Hetherington, H. Shu, and K. Livescu, "Pronunciation modeling using a finite-state transducer representation," *Speech Communication*, vol. 46, no. 2, pp. 189–201, 2005.

[9] L. Hetherington, "The MIT finite-state transducer toolkit for speech and language processing," in *Proc. ICSLP*, 2004.

[10] S. Wang, "Using graphone models in automatic speech recognition," Master's thesis, MIT, 2009.

[11] I. Badr, I. McGraw, and J. Glass, "Learning new word pronunciations from spoken examples," in *Proc. INTERSPEECH*, 2010.

[12] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, vol. 17, pp. 137–152, 2003.

[13] V. Z. et al., "Jupiter: A telephone-based conversational interface for weather information," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.

[14] A. Park, T. Hazen, and J. Glass, "Automatic processing of audio lectures for infomation retrieval: vocabulary selection and language modeling," in *Proc. ICASSP*, 2005.

[15] H.-A. Chang and J. R. Glass, "A back-off discriminative acoustic model for automatic speech recognition," in *Proc. INTERSPEECH*, 2009.