Learning New Word Pronunciations from Spoken Examples

Ibrahim Badr, Ian McGraw, and James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{iab02, imcgraw, glass}@mit.edu

Abstract

A lexicon containing explicit mappings between words and pronunciations is an integral part of most automatic speech recognizers (ASRs). While many ASR components can be trained or adapted using data, the lexicon is one of the few that typically remains static until experts make manual changes. This work takes a step towards alleviating the need for manual intervention by integrating a popular grapheme-to-phoneme conversion technique with acoustic examples to automatically learn high-quality baseform pronunciations for unknown words. We explore two models in a Bayesian framework, and discuss their individual advantages and shortcomings. We show that both are able to generate better-than-expert pronunciations with respect to word error rate on an isolated word recognition task.

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

1. Introduction

In many ways, the lexicon remains the Achilles heel of modern automatic speech recognizers (ASRs). Unlike stochastic acoustic and language models that learn the values of their parameters from training data, the baseform pronunciations of words in an ASR vocabulary are typically specified manually (usually along with the basic phoneme inventory itself), and do not change, unless they are tweaked by an expert.

A more desirable solution would be one whereby the basic linguistic units of a language, and the associated lexical pronunciations could be determined automatically from a large amount of speech data. While there has been some research oriented in this direction, there has also been research that addresses an important practical problem having to do with generating pronunciations for new words. One common approach is to use some form of letter-to-sound generation model to predict pronunciations of new words [1, 2, 3]. An extension of this idea is to incorporate spoken examples to refine the pronunciation [4, 5, 6, 7].

In this work, we also explore the use of spoken examples of new words to improve upon the pronunciation generated by an initial letter-to-sound model. Our research differs from previous work in the stochastic learning framework. In our case, we use an n-gram graphone-based model as the basis for our initial estimate of a pronunciation [3]. The graphone model is used as a form of prior to condition our expectation of possible pronunciations of a new word, given its spelling. We then use spoken examples to further refine the pronunciations, and explore two different stochastic pronunciation models: the first cascades all the examples to find a single best pronunciation, while the second creates a pronunciation mixture model (PMM) to consider multiple pronunciations. We compare both approaches on the telephone-based PhoneBook corpus of isolated words and find that they are able to recover expert-level pronunciation baseforms with relatively few example utterances.

In order to show the inherent robustness of the parallel PMM approach, we also collect a noisy set of Internet-based spoken examples using a crowdsourced data recording method. Despite the acoustic mismatch and varied quality of this corpus compared to the clean PhoneBook speech, we observe that pronunciations generated from the PMM formulation are still able to achieve a significant reduction in word error rate (WER) over manually created baseforms.

2. Grapheme-to-Phoneme Conversion

Following the work of [1, 3], we construct an n-gram model over graphone 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, or baseforms $\mathcal{B}$. A joint model of the letter-to-sound task can be formalized as:

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\mathbf{w}, \mathbf{b}) \tag{1}$$

A graphone, $g = (w, b) \in \mathcal{G} \subseteq (\mathcal{W} \cup \{\epsilon\}) \times (\mathcal{B} \cup \{\epsilon\})$, is a subword unit that maps a grapheme subsequence, w, to a phoneme subsequence, b. In this work, we restrict our attention to singular graphones, in which a mapping is made between at most one grapheme and at most one phoneme (omitting the ϵ to ϵ mapping). Taken together, a sequence of graphones, $\mathbf{g}$, 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 graphone sequences $\mathbf{g} \in S(\mathbf{w}, \mathbf{b})$. The following table shows two possible graphone 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/ϵ	p/p	ϵ/ax	l/l	e/ϵ
$\mathbf{g}_2$	=	c/k	o/ϵ	u/ah	p/p	ϵ/ax	l/l	e/ϵ

Given this ambiguity, employing graphones 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].

$$P(\mathbf{w}, \mathbf{b}) = \sum_{\mathbf{g}\in S(\mathbf{w},\mathbf{b})} P(\mathbf{g}) \approx \max_{\mathbf{g}\in S(\mathbf{w},\mathbf{b})} P(\mathbf{g}) \tag{2}$$

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-pronunciation pairs to automatically infer graphone alignments. We then train a standard 5-gram language model over the automatically segmented corpus of graphones. This configuration has been shown to produce good results for singular graphones [8].

3. Graphone-guided Phonetic Recognition

We begin by exploring a model which incorporates one example utterance with the graphone model to find a single high probability baseform. Given a word or grapheme sequence $\mathbf{w}$ and an example utterance, u, of $\mathbf{w}$ we deduce the baseform for $\mathbf{b}^*$ using a similar framework to that described in [4].

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\mathbf{b}|\mathbf{w}, u) = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\mathbf{b}, \mathbf{w}) p(u|\mathbf{b}, \mathbf{w}) \quad (3)$$

We replace the decision tree originally described in [4] with a graphone n-gram model. For each word, $\mathbf{w}$, a recognizer, $R_{\mathbf{w}}$, can be constructed using weighted finite-state-transducers (FSTs) to model the mapping of acoustic model labels to phoneme sequences, weighted by graphone language model parameters. Given an FST, C, to map context-dependent models to phones, and an FST, P, of phonetic rules to map phones to phonemes, we construct $R_{\mathbf{w}} = C \circ P \circ PLM_{\mathbf{w}}$, where $PLM_{\mathbf{w}}$ is a phoneme language model created by transforming graphones to phones for both the input and output of a $\mathbf{w}$-constrained graphone language model. Decoding during recognition of the single example utterance can be performed using a forward Viterbi search and a backward A^* search.

The procedure described above only incorporates a single example utterance into the pronunciation generation framework. The following sections introduce two methods of utilizing a *set* of M example utterances, u_1^M, of a given word, $\mathbf{w}$.

3.1. Cascading Recognizers

As in equation 3, we apply Bayes rule with the additional assumption of independence between example utterances given their pronunciations to model the probability of a baseform given the data:

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\mathbf{b}, \mathbf{w}) \prod_{i=1}^{M} p(u_i|\mathbf{b}, \mathbf{w})$$

This multiple utterance recognizer can be implemented as a cascade of single utterance recognizers. Recognition can be conceptualized as composing acoustic information, A, with $R_{\mathbf{w}}$ for each utterance to produce a hypothesis lattice U. These lattices can also be represented as FSTs, and projecting their outputs to the inputs (denoted $[\ldots]_{oo}$) allows us to effectively multiply in a subsequent $p(u_i|\mathbf{b}, \mathbf{w})$ term:

$$U_1 = A_1 \circ R_{\mathbf{w}} \quad \text{and} \quad U_i = A_i \circ [U_{i-1}]_{oo}$$

In practice, this formulation introduces concerns regarding the path-pruning performed by the beam-search in the recognizer, but we defer the discussion of this phenomenon for the moment.

3.2. Pronunciation Mixture Model

A second formulation of pronunciation generation informed by multiple example utterances is that of a *pronunciation mixture model* (PMM). We parameterize our model with $\theta_{\mathbf{b},\mathbf{w}} = P(\mathbf{b}, \mathbf{w})$ under the assumption that a particular word $\mathbf{w}$ and baseform $\mathbf{b}$ have some joint probability, however small, of mapping to one another. In a setup similar to the work described in [6], the EM algorithm is used to update these parameters based on the data $(u_1^M, \mathbf{w})$. Whereas Li et al. optimize graphone language model parameters, our goal here is to directly learn weights for word pronunciations, hence the PMM characterization. We begin by characterizing the log-likelihood of the data.

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

The parameters, θ, are initialized to our graphone n-gram model scores and multiple iterations of the EM algorithm are run. The following equations specify the expectation and maximization steps respectively:

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

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

Although in principle we could apply these weights in a stochastic lexicon, for this work we simply pick the baseform(s) $\mathbf{b}$ with the highest probability as the pronunciation of $\mathbf{w}$.

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} P(\mathbf{w}, \mathbf{b}; \theta^*) = \arg\max_{\mathbf{b}\in\mathcal{B}} \theta^*_{\mathbf{w},\mathbf{b}} \quad (4)$$

4. Experimental Setup

To experiment with the two pronunciation models, we use a landmark-based speech recognizer [9]. MFCC averages are computed over varying durations around hypothesized acoustic-phonetic landmarks to generate 112-dimensional feature vectors, which 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 a separate corpus of telephone speech. The search space in the recognizer is modeled using a flexible weighted FST toolkit [10].

The pronunciation models were evaluated on the task of isolated word recognition using the PhoneBook corpus [11]. To ensure adequate data for our baseline experiments, we chose a random 2,000 word subset that each had example spoken utterances from at least 13 distinct speakers. We also ensured that expert pronunciations existed in our lexicon. We held out two of the 13 utterances, one from a male speaker and the other from a female speaker, to generate a 4,000 utterance set.

While the individual recognition experiments described in the next section are limited to the 2,000 selected words, a far larger lexicon was used to train the initial graphone language model parameters. For this work we used an internal dictionary that contains over 150,000 manually generated entries. To simulate the out-of-vocabulary scenario for which graphones are typically employed, we removed the 2,000 trial words from our lexicon, and further pruned similarly spelled words using a simple edit distance criterion. We then trained a 5-gram graphone language model according to the procedures described in [8].

We conducted two baseline experiments to frame our remaining results. The first was a graphone-only baseline in which we performed isolated word recognition over the 4,000 test utterances using the 2,000 word pronunciation lexicon generated from the graphone model alone according to equation 2. Since no acoustic information was used, this provided us with an initial unsupervised WER of 16.7%. The second baseline was again the 2,000 word-recognition task; however, this time we explicitly used the manually generated pronunciations originally found in our lexicon, giving us a target WER of 12.4%, achievable directly by experts.

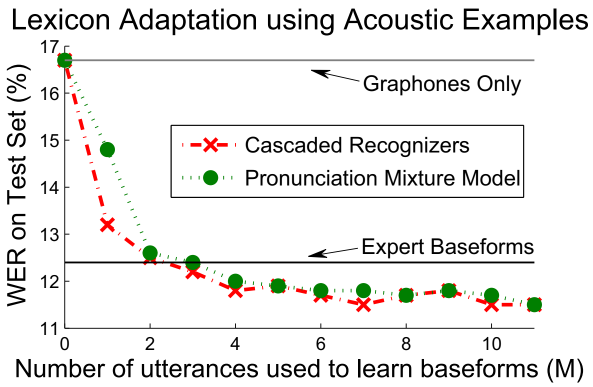

Figure 1: Word Error Rate (WER) as a function of the number of example utterances used to adapt the underlying lexicon.

It should be noted that about 160 words in the expert-lexicon had multiple baseforms associated with them. For example, the word "youths" was represented as both *y uw dh z* and *y uw th s*. Initial experiments indicated that allowing multiple baseforms could give an advantage to the expert-lexicon that could be leveraged in the other frameworks. We begin however by choosing only a single pronunciation for inclusion in an automatically generated lexicon. Even so, were able to show the feasibility of recovering and even surpassing the performance of manually generated baseforms.

5. Experimental Results

Having established our baseline experiments, we evaluated both the cascading recognizer approach and the PMM by varying the number of training utterances for each, and evaluating the WER of the test set against the lexicons produced under each condition. The resulting plot is shown in figure 1. It is encouraging to note that both models perform admirably, achieving expert-level pronunciations with just three example utterances.

5.1. Cascading Recognizers

The cascading recognizer approach of section 3.1 improves slightly faster than the PMM technique. With seven utterances, this model surpasses the expert baseform WER by nearly 1%.

An inherent assumption of this model is that there is a single, correct underlying pronunciation. This fact may explain the slight advantage that this approach has, since our experimental design only allows a single baseform for each word in our automatically generated lexicon. A model which directly computes the single most likely baseform given the data is thus particularly well-suited to the task.

Ideally, a pronunciation generation model would be able to cope with words that have multiple pronunciations, such as "either". It probably does not make sense, for example, to be multiplying the acoustic scores of one utterance pronounced *iy dh er* with a second pronounced *ay dh er*.

Lastly, another potential pitfall of this approach is that unless special care is taken to customize the pruning procedure, acoustic variation will inherently cause the pronunciation search space to become successively smaller as the compositions prune low-probability paths. This is especially problematic when considering noisy utterances. Indeed, even with the clean speech comprising the PhoneBook corpus, by the 11th utterance, N-best lists produced by the cascaded recognizers contained an average of just 10.7 entries.

τ	0.2	0.1	0.05
Avg. # **b** per **w**	1.25	2.10	2.21
WER (%)	11.2	**11.0**	11.5

Table 1: By varying a threshold τ over the weights learned in the PMM, we can incorporate multiple baseform pronunciations for individual words.

5.2. Pronunciation Mixture Model

To illustrate the performance of the PMM, we plot in figure 1 the WER obtained by generating a lexicon according to equation 4 after two iterations of EM. This stopping criterion was determined by constructing a development set of 1,500 previously discarded PhoneBook utterances and running recognition using lexicons generated after each EM iteration. Alternatively, EM could have been run to convergence and then smoothed, again with the aid of a development set.

While the PMM requires slightly more data to achieve the lowest reported WER of the cascade approach (11.5%), it is eventually able to do so once all 11 training utterance are incorporated into the mix. It is clear from the figure that with only a single training example EM begins to over-fit the acoustic idiosyncrasies of that particular example. Though not shown in the figure, this effect is magnified for small amounts of training data when EM is run for a third and fourth iteration.

One advantage of the PMM approach is that it directly models multiple pronunciations for a single word, an avenue we begin to explore with a second set of preliminary experiments. We use a simple weight threshold $\theta_{\mathbf{w},\mathbf{b}} > \tau$, to choose baseforms for inclusion. As in the single baseform case, we discard the weights once the baseforms have been chosen, but we ultimately envision them being utilized during decoding in a stochastic lexicon.

Table 1 shows WER obtained by recognizers with lexicons generated under varying values of τ. Choosing $\tau = 0.1$ yields the best reported WER of 11.0%, a 1.4% absolute improvement over the expert-baseline. It's interesting to note that this threshold implies an average of 2.1 pronunciations per word, almost double that of the expert lexicon which has 1.08.

5.3. Noisy Acoustic Examples

Models that incorporate acoustic information into lexicon adaptation become particularly useful in domains where acoustic data is cheaper to obtain that expert input. In [6], example utterances of spoken names are obtained in an unsupervised fashion for a voice-dialing application by filtering for interactions where the user confirmed that a call should be placed. Unfortunately, not all domains are amenable to such a convenient context-filter to find self-labeled utterances.

To collect arbitrary acoustic data, we turned to the Amazon Mechanical Turk (AMT) cloud-service. AMT has been described as a work-force in the cloud since it enables *requesters* to post web-based tasks to any *workers* willing to accept micro-payments of as little as \$0.005 upon completion. The service has become popular in the natural language processing community for collecting and annotating corpora, and has recently been gaining use in the speech community. In [12], we were able to collect over 100 hours of read speech, in under four days.

In this work, we used a similar procedure to augment our PhoneBook corpus with another 10 example utterances for each of its 2,000 words at a cost of \$0.01 per utterance. Whereas in [12] we took care to filter the collected speech to obtain high-quality sub-corpora, we took no such precautions when collect-

	# Utts.	Iter.1	2	3	4
Phonebook	11	12.3	11.5	11.7	12.0
AMT	10	12.3	**12.0**	13.0	15.3
Phonebook+AMT	21	12.3	11.6	11.6	12.0

Table 2: PMM results incorporating spoken examples collected via Amazon Mechanical Turk.

Word	Dictionary Baseform	Top PMM Baseform
parishoners [sic]	p **AE** r ih sh ax n er z	p **AX** r ih sh ax n er z
traumatic	tr r **AO** m ae tf ax kd	tr r **AX** m ae tf ax kd
winnifred	w ih n ax f r **AX** dd	w ih n ax f r **EH** dd
crosby	k r ao **Z** b iy	k r aa **S** b iy
melrose	m eh l r ow **Z**	m eh l r ow **S**
arenas	**ER** iy n ax z	**AX R** iy n ax z
billowy	b ih l **OW** iy	b ih l **AX W** iy
whitener	w ay **TF AX** n er	w ay **TD** n er
airsickness	eh r **SH** ih kd n **EH** s	eh r **S** ih kd n **AX** s
Isabel	**AX S AA** b eh l	**IH Z AX** b eh l

Table 3: Example baseform changes between expert dictionary and top PMM hypothesis. Phonemes involved in the difference have been capitalized.

ing these example utterances. Thus, in addition to other sources of mismatch between the data and our acoustic model, this noisy data poses a challenge to even a recognizer built on expert pronunciations. Running the expert baseline recognizer over these 20,000 utterances yields a very high WER of 50.1%. Of course, since we could make no guarantees that the worker even read the word, the true error rate is unknown.

It might seem, then, that using this data to generate valid pronunciations is a dubious exercise. Indeed, this data set confounds the cascading recognizer configuration since a single noisy utterance can throw off the entire cascade. Fortunately, the PMM approach has the nice property that a few noisy scores do not significantly affect the totals.

Repeating a subset of the experiments of the previous section, we again show four iterations of the PMM approach, using the PhoneBook utterances alone, AMT-PhoneBook combined utterances, and the AMT-collected corpus alone. Despite the noisy nature of the cloud-collected corpus, table 2 shows that there is little degradation in WER when using all 21 utterances for every word. Perhaps more pleasing is the fact that generating pronunciations based on just the AMT-data still manages to out-perform even the expert generated pronunciations, achieving a WER of 12.0% compared with 12.4% for the experts.

5.4. Analysis of Learned Baseforms

In order to quantify some of the differences between the expert and learned baseforms, we ran NIST align software to tabulate differences between the reference expert baseform, and the top choice hypothesis of the PMM model. Of the 2000 baseform pairs, 83% were identical, while the remainder mostly contained a single substitution. Most of the substitutions involved vowels, typically a schwa. Only 2% of the data contained an additional insertion or deletion. Most of these involved retroflexed vowel sequences.

Table 5.4 shows examples of common confusions including vowel and consonant substitutions, vowel/semi-vowel sequence perturbations, syllable deletions, and outright pronunciation corrections. Although the latter were few, it was encouraging to see that they did occur.

6. Summary and Future Work

This work has introduced and compared two promising approaches to generating pronunciations by combining graphone techniques with acoustic examples. Furthermore, we have shown that even in the presence of significant noise, a pronunciation mixture model can reliably generate improved baseform pronunciations over those generated by experts.

The improvements we have observed by allowing multiple pronunciations in our lexicon suggests two other avenues of exploration. First, we might try to incorporate the weights learned by a PMM directly into a stochastic lexicon. Second, rather than relying on pronunciation rules to govern the mapping between phonemes and phones, we might try to learn lexicon mixture entries directly at the phonetic level.

Another area ripe for exploration is the joint training of lexical baseforms and the acoustic model. A first experiment might hold each component fixed, while training the other, in hopes of converging on a lexicon consistent with the acoustic models which are in turn directly optimized for the target domain.

If these initial results extend to other domains, the possibility of learning better-than-expert baseforms in arbitrary domains opens up many possibilities for future work. For example, when faced with an out-of-vocabulary word with a known spelling, any system could programmatically post a task to AMT and collect example utterances to generate a high quality entry in the lexicon.

Long term, the ultimate goal of this research might be to learn pronunciations from entirely flat language models over sub-word units. If it were feasible to simultaneously train the lexicon, acoustic model, *and* sub-word language model from scratch, large vocabulary speech recognizers could be built for many different languages with little to no expert input, if given enough orthographically transcribed data.

7. 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, "Automatic baseform generation from acoustic data," in *Proc. ASRU*, 2007.

[7] G. F. Choueiter, M. I. Ohannessian, S. Seneff, and J. R. Glass, "A turbo-style algorithm for lexical baseforms estimation," in *Proc. ICASSP*, 2008.

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

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

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

[11] J. Pitrelli and C. Fong, "Phonebook: NYNEX isolated words," `http://www.ldc.upenn.edu`.

[12] I. McGraw, C. Lee, L. Hetherington, S. Seneff, and J. Glass, "Collecting voices from the cloud," in *Proc. LREC*, 2010.