ADAPTING GRAPHEME-TO-PHONEME CONVERSION FOR NAME RECOGNITION

Xiao Li, Asela Gunawardana and Alex Acero

Microsoft Research
One Microsoft Way, Redmond, WA, 98052

ABSTRACT

This work investigates the use of acoustic data to improve grapheme-to-phoneme conversion for name recognition. We introduce a joint model of acoustics and graphonemes, and present two approaches, maximum likelihood training and discriminative training, in adapting graphoneme model parameters. Experiments on a large-scale voice-dialing system show that the maximum likelihood approach yields a relative 7% reduction in SER compared to the best baseline result we obtained without leveraging acoustic data, while discriminative training enlarges the SER reduction to 12%.

Index Terms— grapheme-to-phoneme conversion, pronunciation model, name recognition, discriminative training

1. INTRODUCTION

Grapheme-to-phoneme (G2P) conversion, sometimes referred to as letter-to-sound conversion, has become an indispensable component in large-scale voice-dialing systems. Many state-of-the-art G2P conversion systems are based on statistical models [1, 2, 3, 4], where probabilistic relationships between graphemes and phonemes are learned from a hand-authored pronunciation lexicon consisting of common English words. [1] A G2P model trained in such a manner, however, may or may not be optimal when applied to large-scale name recognition tasks (with over 10^4 names). The first challenge comes from domain mismatch; some grapheme-phoneme relationships that occur in names may be lacking from a pronunciation lexicon. Although we can reduce this mismatch by adding names and their pronunciations into the lexicon, it is unrealistic to do it at a large scale, as the number of unique names can be gigantic, and it is often the rare names that have irregular pronunciations. The second challenge is speaker variability. People from different geographic regions and ethnic groups may pronounce the same name in different ways; and a hand-authored pronunciation lexicon can hardly capture all such variations.

Ideally, G2P conversion should produce pronunciations that best serve the target application, which in our case is name recognition. To this end, we propose to leverage acoustic data obtained from a large-scale voice-dialing system to adapt a G2P model for name recognition. In fact, there has been various work on learning pronunciations for proper names from acoustic data [5, 6, 7]. A common goal thereof is to directly augment or modify an existing pronunciation lexicon with pronunciations generated from acoustic data. A key difference of our work is that we aim at adapting at the graphoneme level, which will be introduced shortly. Theoretically speaking, given sufficient adaptation data, the resulting G2P conversion should not only improve pronunciation for those words that have occurred in adaptation data, but also generalize to unseen words.

The rest of the paper is organized as follows: Section 2 reviews a graphoneme ngram model for grapheme-to-phoneme conversion. Section 3 introduces a joint model of acoustics and graphonemes. Section 4 and Section 5 respectively present maximum likelihood training and discriminative training for G2P model adaptation. Section 6 discusses how to obtain adaptation data in an unsupervised manner. Section 7 presents name recognition experiments and results, and Section 8 concludes.

2. GRAPHONEME NGRAM MODELS

In this work, we construct probabilistic relationships between graphemes and phonemes using a graphoneme ngram model (also referred to as a joint multi-gram model) [2, 3]. For readers' convenience, we briefly review how we create graphoneme sequences from a pronunciation lexicon, based on which a graphoneme ngram model can be trained.

grapheme seq.	l	e	t	t	e	r
phoneme seq.	l	eh	t	ϵ	ax	r
graphoneme seq.	l:l	e:eh	t:t	t:ϵ	e:ax	r:r

Table 1. An example of a graphoneme sequence

We let a random variable g denote a grapheme sequence and let ϕ denote a phoneme sequence. Furthermore, we use s to represent an alignment *and* a grouping of ϕ and g, as will be defined in the following example. Consider the word

[1] This paper studies G2P conversion for the language of English, though the ideas presented here can potentially be applied to other languages.

978-1-4244-1746-9/07/$25.00 ©2007 IEEE

letter, which has g = (l, e, t, t, e, r) and ϕ = (l, eh, t, ax, r). One possible way of aligning g and ϕ is shown in Table 1, where ϵ denotes a *null* phoneme. Given such an alignment, primitive graphoneme units can be generated by associating graphemes with their phoneme counterparts, as shown in the last row of Table 1. Next, adjacent graphoneme units can be grouped together to form larger units. In the above example, merging l:l with e:eh, and e:ax with r:r, results in

$$\text{l\&e:l\&eh} \quad \text{t\&t:t\&}\,\epsilon \quad \text{e\&r:ax\&r} \tag{1}$$

The form of (1) is what we define a graphoneme sequence, which is fully determined by (g, ϕ, s).

Having introduced the concept of graphonemes, we now turn to the question of how to create such graphoneme sequences as in (1) from a pronunciation lexicon of parallel grapheme and phoneme sequences, *i.e.*, how to infer s given a set of (g, ϕ) pairs. The first step is to automatically align g and ϕ to form primitive graphonemes. This work adopts an EM approach presented in [2], where alignment is inferred using graphoneme unigram statistics. Secondly, we follow a procedure similar to [3] to merge graphemes into larger units except that our algorithm is based on mutual information instead of co-occurring frequency, and that we allow a graphoneme unit to have maximally k graphemes and l phonemes.

Once we create a corpus of graphoneme sequences, we train a standard ngram model with backoff. Depending on the amount of training data, we use a cutoff threshold to adjust model complexity; an ngram will be excluded from the model if it has a count no more than this threshold. Finally, G2P conversion can be achieved by applying best-first search (or other search algorithms) [3]. Details about training/decoding of a graphoneme ngram model can be found in [2, 3]. Here we focus our attention on the use of acoustics in adapting such a model, which we will present next.

3. A JOINT MODEL OF ACOUSTICS AND GRAPHONEMES

As mentioned in the introduction, the end-to-end goal of this work is to optimize G2P conversion to improve name recognition. In this regard, acoustic data can be very useful in learning grapheme-phoneme relationships that occur in real-world applications. We introduce another random variable x to represent acoustics, and we propose to jointly model x, g, ϕ and s as follows,

$$\log p_\theta(x, g, \phi, s) = \log p(x|\phi) + \log p_\theta(g, \phi, s) \tag{2}$$

The factorization follows the assumption that x is independent of g and s given ϕ. Therein, the joint likelihood is expressed by an acoustic model score $p(x|\phi)$ and a graphoneme model score $p_\theta(g, \phi, s)$, where θ represents ngram model parameters to be adapted. Note that we use a fixed acoustic model, and $p(x|\phi)$ is therefore not parameterized. Moreover, we add a scale factor a which serves similarly to a language model scale factor in speech recognition; Equation (2) hence becomes

$$\approx \log p(x|\phi) + a \log p_\theta(g, \phi, s) \tag{3}$$

For simplicity, we omit a in all our following formulation, but keep in mind that a is applied to Equation (2) in practice.

Moreover, we assume that both x and g are observable, whereas ϕ and s are hidden. We specifically assume the availability of a set of *adaptation data* (x_i, g_i). In Section 6, we will describe how we obtain grapheme labels g for acoustic data x in an unsupervised manner. Given the labeled data, one potential approach to adapting a graphoneme ngram model is to re-estimate model parameters that maximize the joint likelihood $\log p(x, g)$, leading to maximum likelihood estimation (MLE). Alternatively, we can directly maximize the conditional likelihood $\log p(g|x)$ using a discriminative training (DT) approach. We will discuss these two approaches respectively in the following two sections.

4. MAXIMUM LIKELIHOOD TRAINING

4.1. Maximizing joint likelihood

Given a set of (x_i, g_i) pairs, the objective of MLE is to maximize

$$\sum_{i=1}^{m} \log p_\theta(x_i, g_i) = \sum_{i=1}^{m} \log \sum_{\phi_i, s_i} p_\theta(x_i, g_i, \phi_i, s_i) \tag{4}$$

Standard EM algorithm can be applied to cope with hidden variables $\{\phi_i, s_i\}_{i=1}^{m}$. Alternatively, we can apply the Viterbi algorithm, which we adopt in this work for simplicity. The special optimization procedure is as follows,

1. Start from a baseline graphoneme model θ_0 that is trained on a pronunciation lexicon (see Section 2).

2. Find the most likely ϕ_i and s_i, given the observed $(x_i$, $g_i)$, and the current model estimate θ, *i.e.*

$$\begin{aligned} \hat{\phi}_i, \hat{s}_i &= \underset{\phi_i, s_i}{\text{argmax}} \ \log p_\theta(\phi_i, s_i | x_i, g_i) \\ &= \underset{\phi_i, s_i}{\text{argmax}} \ \log p(x_i|\phi_i) + \log p_\theta(g_i, \phi_i, s_i) \end{aligned} \tag{5}$$

3. Re-estimate the model by

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \sum_{i=1}^{m} \log p_\theta(g_i, \hat{\phi}_i, \hat{s}_i) \tag{6}$$

4. Repeat step 2 and 3 until convergence.

For computational convenience, for each i, the "argmax" operation in Equation (5) is taken only w.r.t. the top n phoneme sequences ϕ_i that yield the highest $\log p_\theta(g_i, \phi_i, s_i)$ scores.

In other words, we use the current model to generate n-best phoneme sequences and then use Equation (5) to rescore them. This is akin to the idea of combining linguistic knowledge and acoustic data to generate pronunciation lexicons [6, 7]. Note that the n-best list could also be generated by a phonetic decoder, but this would introduce "linguistically incorrect" pronunciations that are not desired in training a G2P model.

Another issue worth attention is when g_i is not the correct label for x_i (as will be explained in Section 6). We need to discard such noisy samples which would otherwise "contaminate" the graphoneme model. A simple method is to use an acoustic model confidence α; a sample is discarded if

$$\log p(x_i|\hat{\phi}_i) < \alpha \tag{7}$$

The intuition is that, if g_i is not the correct label for x_i, then it is unlikely that any of the n-best ϕ_i (and hence $\hat{\phi}$) would yield a high acoustic model score.

4.2. Adaptation strategies

The above approach yields a graphoneme model that is optimized w.r.t. an adaptation set. Depending on the amount of adaptation data, this model may or may not generalize well. A more robust approach would be to leverage information from the pronunciation lexicon on which the baseline graphoneme model is trained. This resembles the idea of language model adaptation which attempts to learn models for a new domain (often with limited data) by leveraging existing, out-of-domain data [8]. In this work, we investigate two simple strategies in the context of adapting a graphoneme model.

Model interpolation: we obtain model θ^{ML} from Equation (6) (after convergence), and interpolate it linearly with the baseline graphoneme model θ_0. The interpolation weights are tuned on a development set.

Data combination: we obtain $\hat{\phi}_i$ from Equation (5) (again after convergence) for each i. Then we combine $\{(g_i, \hat{\phi}_i)\}_{i=1}^m$ with the original pronunciation lexicon, and retrain a model following Section 2. In this regard, $\{(g_i, \hat{\phi}_i)\}_{i=1}^m$ functions like a "pronunciation lexicon" that is generated from acoustic data. However, unlike a typical pronunciation lexicon where each (g, ϕ) value is unique, $\{(g_i, \hat{\phi}_i)\}_{i=1}^m$ can contain identical entries, *i.e.* $(g_i = g, \hat{\phi}_i = \phi)$ for multiple i. In fact, this redundancy can be useful to our task, as it naturally defines a prior distribution $p(g, \phi)$ that is absent from a pronunciation lexicon. To support our argument, we will conduct an experiment in which we remove this redundancy by collapsing identical entries after data combination.

We formally evaluate these adaptation strategies in Section 7, where we use adapted graphoneme models, instead of the baseline model, to generate pronunciations for in-grammar names, and where we measure recognition error rates on a test set.

5. DISCRIMINATIVE TRAINING AND RESCORING

MLE aims to find parameters that best describe the data, and is statistically consistent under the assumptions that the model structure is correct, that the training data is generated from the true distribution, and that we have an infinite amount of such training data. Such conditions, however, are rarely satisfied in practice. Discriminative training (DT), which directly targets for better classification/recognition performance, often yields superior performance.

5.1. Maximizing conditional likelihood

In the context of grapheme-phoneme conversion, the goal of DT is to estimate graphoneme model parameters in such a way that pronunciations generated by this model maximally reduce recognition error. The goal is similar to DT of language models for speech recognition [9]. In this work, we maximize the conditional likelihood of a grapheme sequence given acoustics, *i.e.*

$$\sum_{i=1}^{m} \log p_\theta(g_i|x_i) = \sum_{i=1}^{m} \log \frac{p_\theta(x_i, g_i)}{\sum_{g_i'} p_\theta(x_i, g_i')} \tag{8}$$

The computation of $p(x_i, g_i)$ involves the marginalization over ϕ_i, s_i . Here we make the approximation that

$$p_\theta(x_i, g_i) = \sum_{\phi_i, s_i} p_\theta(x_i, g_i) \approx p_\theta(x_i, g_i, \hat{\phi}_i, \hat{s}_i) \tag{9}$$

where $\hat{\phi}_i, \hat{s}_i$ are defined in Equation (5). Equation (8) consequently becomes

$$\approx \sum_{i=1}^{m} \log \frac{p(x_i|\hat{\phi}_i) p_\theta(\hat{\phi}_i, \hat{s}_i, g_i)}{\sum_{g_i'} p(x_i|\hat{\phi}_i) p_\theta(\hat{\phi}_i', \hat{s}_i', g_i')} \tag{10}$$

Stochastic gradient descent [10] can be applied to find a locally optimal estimate θ^{DT}.

Specifically, the training procedure is carried out as follows:

1. Start with an ML-adapted graphoneme model θ^{ML} (Section 4);
2. For x_i, obtain n-best recognition results g_i' using a speech recognizer and using the ML-adapted graphoneme model;
3. For (x_i, g_i), obtain $\hat{\phi}_i, \hat{s}_i$ by Equation (5); and similarly for each (x_i, g_i'), obtain $\hat{\phi}_i', \hat{s}_i'$ by Equation (5);
4. Apply stochastic gradient descent to Equation (10) w.r.t. θ; apply early stopping [11] to avoid overfitting.
5. Repeat step 2, 3 and 4 until convergence.

There is one issue we would like to address before moving on. In an ngram model with backoff, if we encounter an ngram that does not exist in the model, we compute its probability by backing off to a lower-order distribution. There are several options regarding how to handle backoff in DT [9]. In this work, we choose to fix backoff weights while updating lower-order ngram parameters.

5.2. Rescoring

For consistency with training in which optimization is conducted on n-best grapheme sequences (see Equation (10)), we evaluate the discriminatively trained model in a similar fashion. For each x_i in the test set, we generate n-best g'_i using a speech recognizer and using the ML-adapted graphoneme model θ^{ML}. Then we rescore g'_i using the model obtained by discriminative training.

$$\begin{aligned}\hat{g}_i &= \underset{g'_i}{\operatorname{argmax}}\, p_{\theta^*}(g'_i|x_i) \\ &= \underset{g'_i}{\operatorname{argmax}}\, p(x_i|\hat{\phi}'_i)p_{\theta^{DT}}(\hat{\phi}'_i, \hat{s}'_i, g'_i)\end{aligned} \tag{11}$$

Here we make the same approximation as we did in Equation (10). Finally, we measure recognition error rates based on $\hat{g}_i$ obtained from rescoring.

6. ADAPTATION DATA ACQUISITION

Our discussion so far assumes the grapheme labels of acoustic data are available at adaptation time. Manual transcription in a large-scale voice dialing system is an expensive and error-prone task due to the large number of (and sometimes confusable) names in the grammar. What we propose in this work is to obtain grapheme labels for a subset of acoustic data by dialog analysis. Specifically, we utilize data in "successful" dialog sessions. Here by "successful", we mean that the dialog ends up with an automatic transfer (to the person of interest) after a positive confirmation from the user, *e.g.*,

System: "Good morning. Who would you like to contact?"
User: "John Doe."
System: "Did you say John Doe?" (generated by TTS)
User: "Yes."
System: "O.K. I'll transfer you in a moment."

At the end of this dialog session, the system will log an event that the call was transferred to *John Doe*. Since the user confirmed "yes" to the system before the transfer, it is reasonable to assume that the name of the person to whom the call is transferred is the correct grapheme label for the corresponding waveform. Sometimes, the system may go through multiple rounds of interactions before the user gives a positive confirmation, then the grapheme label obtained from the final transfer may correspond to multiple waveforms in that dialog session.

By making such an assumption, however, we potentially introduce noise in our data — the destination to which a call is transferred may not be the true grapheme label for the corresponding waveform (waveforms). In fact, we have observed instances where a user confirmed "yes" to the system even he/she was recognized wrong (often due to confusable pronunciations generated by TTS), and the call got transferred to a wrong person in the end. This is the main reason we apply Equation (7) to remove noisy data from the adaptation set.

7. EVALUATION

This section examines whether a G2P conversion system using adapted graphoneme models would improve name recognition performance. For a fair comparison between different graphoneme models, we disabled the pronunciation lexicon lookup in our speech recognizer. In other words, all pronunciations must be obtained via G2P conversion. In addition, we set the graphoneme model scale factor $\alpha = 0.25$ in all cases, which empirically worked well.

7.1. Data sets

We have two professionally transcribed pronunciation lexicons containing (g, ϕ) pairs. The first one is a "general lexicon" with common English words, including frequently used names (some are foreign names). This lexicon has 81K pronunciations for 65K unique words, where each word can have multiple pronunciations. The other lexicon, which we call a "name lexicon", contains solely names. It has 64K pronunciations for 53K unique words. These two lexicons are to be used in training baseline graphoneme models.

Furthermore, following the procedure described in Section 6, we obtained adaptation data, *i.e.*, (x, g) pairs, from the call logs of a corporate voice-dialing system with 58K names. We collected an adaptation set with 30K utterances of *first name + last name*. We also created a second adaptation set (from a different period when the calls were made) to serve as a development set.

Finally, we prepared a separate test set from the same voice-dialing system. This test set contains 2844 utterances and 5719 word tokens. It is different from an adaptation set in that the grapheme labels of the waveforms were transcribed by human. In evaluation, these 2844 utterances were tested against a grammar with 58K names — the same grammar as was used in the voice-dialing system.

7.2. Baseline setup

We first trained baseline graphoneme models using pronunciation lexicons as described in Section 2, and evaluated their performance on the test set using Microsoft telephony speech recognition engine. Note that since each utterance in the test set (also in the adaptation set) consists of a first name and a

ID	Description	% WER	% SER
(a)	**General lexicon**	**10.42**	**13.15**
(b)	Name lexicon	11.21	13.99
(c)	General + name	10.70	13.29
(d)	Adapt	10.61	13.40
(e)	(a) & (d) interpolated	9.86	12.48
(f)	**General + adapt**	**9.58**	**12.20**
(g)	General + adapt collapsed	10.16	12.83
(h)	General + increased adapt	9.65	12.31

Table 2. First-pass recognition results using baseline and MLE models (graphoneme trigrams without cutoff); see detailed experiment descriptions in the text.

last name, we measure both word error rate (WER) and sentence error rate (SER). We empirically experimented with different graphoneme unit sizes, ngram orders and ngram cutoff thresholds. The best recognition performance was achieved when we allow *maximally* 4 graphemes and 3 phonemes in a graphoneme unit, and when we use trigrams without cutoffs. We report three experiments under this configuration: (a) using the general lexicon only in training; (b) using the name lexicon only; and (c) using the combined lexicon. As shown in the first three rows of Table 2, the graphoneme trigram model trained using the general lexicon outperformed the other two. We thus use this model as the initial point in adaptation.

Data analysis shows that the general lexicon covers 67% of the words (first name *or* last name) in the test set, whereas the name lexicon covers 74%. Despite the fact that the name lexicon better matches our test set, it does not prevail in recognition. This is probably because some name pronunciations provided by linguists do not match those in real-world applications, or at least do not capture enough variations (considering there are only about 1.2 pronunciations per word in the name lexicon).

7.3. MLE results

Next, we conducted a set of maximum likelihood adaptation experiments as presented in Section 4, and we evaluated the following adapted graphoneme models: (d) a model trained by MLE using adaptation data only; (e) the baseline model interpolated with the one trained using adaptation data (model interpolation); (f) a model trained on a combined data set of the general lexicon and adaptation data (data combination); and (g) which is similar to (f) except that identical (g, ϕ) pairs were collapsed into single entries.

As shown in Table 2, the model trained on combined data obtained the largest reduction error rates — a relatively 8.6% reduction in WER and 7.2% in SER. The model interpolation approach performed almost as good. Moreover, a comparison of (e) and (g) indicated that a prior distribution on $p(g, \phi)$, which was naturally modeled by allowing identical entries in the combined data, was indeed helpful.

In addition, we doubled the amount of adaptation data (by adding the development set into the adaptation set). We then repeated the data combination experiment, shown as (h) in Table 2, where we did not observe further improvement.

7.4. DT results

As explained in Section 5, we propose to conduct n-best rescoring, as opposed to first-pass decoding, to evaluate DT performance. Specifically, we initialized the model using the best model obtained by maximum likelihood adaptation, *i.e.* (f) in Table 2, then we applied DT on the adaptation set. In Figure 1 and Figure 2, we compare WERs and SERs of the baseline graphoneme model trained on the general lexicon, the best ML-adapted model (trained on combined data), and the discriminatively trained model. Note that the error rates

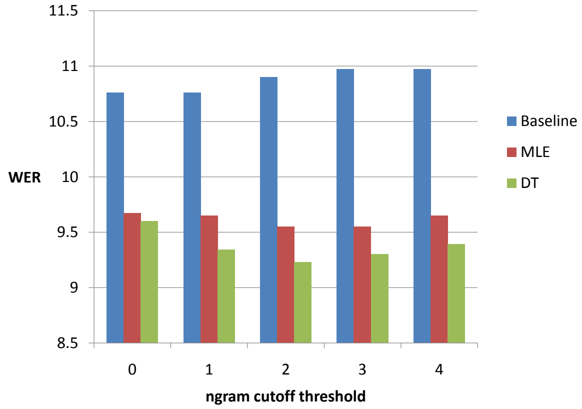

Fig. 1. Rescoring results, in %WER, of the baseline, MLE, and DT models, using trigrams with different cutoffs.

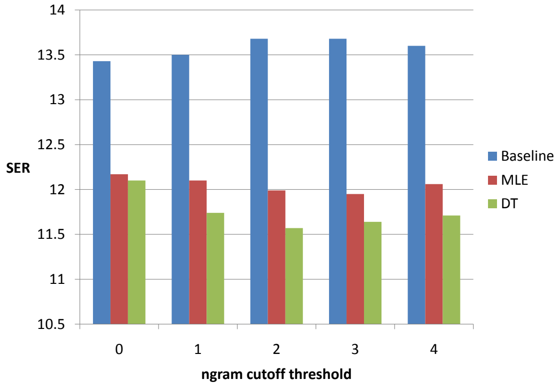

Fig. 2. Rescoring results, in %SER, of the baseline, MLE, and DT models, using trigrams with different cutoffs.

of the first two models are slightly different from those in Table 2 due to the difference between rescoring and first-pass decoding. Furthermore, we tried a number of ngram cutoff thresholds and found that moderately increasing the threshold slightly improved DT performance. This is probably because more aggressive cutoffs would reduce model complexity, thus preventing the model from overfitting the adaptation data. When we used a cutoff threshold of 2, the WER/SER of using the DT model are $9.23\%/11.57\%$, a relative $11.5\%/11.9\%$ reduction from our best baseline from first-pass decoding (the baseline from rescoring performed worse).

Although DT of a graphoneme model significantly improved G2P conversion as measured by recognition performance, we need to be careful when applying a discriminatively trained model in real-world systems. The caveat is that such a model is no longer optimal once the grammar changes — maximal discrimination on one set of names does not mean the same on another. Therefore, the model needs to be retrained if the grammar is significantly changed. The MLE approach, on the other hand, does not have this problem since it aims at maximizing the joint likelihood, which is essentially the numerator in Equation (8).

8. CONCLUSIONS

This paper presented a framework of leveraging acoustic data in adapting G2P conversion for name recognition. We introduced a joint model of acoustics and graphonemes, where graphoneme parameters can be estimated using a maximum likelihood criterion. We examined several adaptation strategies which attempt to combine information from a pronunciation lexicon and from acoustic data. Experiments showed that the best performance came from a simple data combination strategy, which yielded a relative WER/SER reduction of 8.6%/7.2%. Furthermore, we then applied discriminative training to our best ML-adapted model, enlarging the WER/SER reduction to 11.5%/11.9%.

The authors would like to thank Milind Mahajan, Patrick Nguyen and Mei-Yuh Hwang for useful discussions.

9. REFERENCES

[1] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech*, Madrid, Spain, 1995.

[2] M. Bisani and H. Ney, "Investigations on jointmultigram models for grapheme-to-phoneme conversion," in *Proc. ICSLP*, Denver, U.S.A., 2002.

[3] P. Vozila, J. Adams, Y. Lobacheva, and R. Thomas, "Grapheme to phoneme conversion and dictionary verification using graphonemes," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

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

[5] F. Bechet, R. de Mori, and G. Subsol, "Dynamic generation of proper name pronunciations for directory assistance," in *Proc. ICASSP*, Orlendo, U.S.A, 2002.

[6] F. Beaufays, A. Sankar, S. Williams, and M. Weintraub, "Learning name pronunciations in automatic speech recognition systems," in *Proc. IEEE Intl. Conf. on Tools with Artificial Intelligence*, Sacramento, U.S.A., 2003.

[7] G. Chung, C. Wang, S. Seneff, E. Filisko, and M. Tang, "Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation," in *Proc. ICSLP*, Jeju Island, Korea, 2004.

[8] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *Proc. ICASSP*, 2003.

[9] H.-K. J. Kuo, E. Fosler-Lussier, H. Jiang, and C.-H. Lee, "Discriminative training of language models for speech recognition," in *Proc. ICASSP*, Orlando, U.S.A., 2002.

[10] J. C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, 2003.

[11] M. C. Nelson and W. T. Illingworth, *A Practical Guide to Neural Nets*, Addison-Wesley, 1991.