Speech Recognition without a Lexicon - Bridging the Gap between Graphemic and Phonetic Systems

*David Harwath and James Glass**

MIT Computer Science and Artificial Intelligence Laboratory,
Cambridge, Massachusetts 02139, USA
dharwath@csail.mit.edu, glass@mit.edu

Abstract

Modern speech recognizers rely on three core components: an acoustic model, a language model, and a pronunciation lexicon. In order to expand speech recognition capability to low-resource languages and domains, techniques to peel away the expert knowledge required to craft these three components have been growing in popularity. In this paper, we present a method for automatically learning a weighted pronunciation lexicon in a data-driven fashion without assuming the existence of any phonetic lexicon whatsoever. Given an initial grapheme acoustic model, our method utilizes a novel technique for semi-constrained acoustic unit decoding, which is used to help train a letter to sound (L2S) model. The L2S model is then used in conjunction with a Pronunciation Mixture Model (PMM) to infer a pronunciation lexicon. We evaluate our method on English as well as Lao and Haitian, two low-resource languages featured in the IARPA Babel program.

Index Terms: lexicon learning, pronunciation modeling

10.21437/Interspeech.2014-568

1. Introduction

1.1. Previous Work

The cornerstone of modern day speech recognizers is the pronunciation lexicon, which serves as the link connecting the acoustic model and the language model. The lexicon is typically handcrafted by expert humans, which is a costly and time consuming process. Methods to automatically infer high-quality lexicons not only for under-resourced languages, but also new domains in high-resourced languages, are growing in their appeal. In one recent effort, Lu et al [1] assumed the existence of a small seed lexicon, consisting of handcrafted phonetic pronunciations for several thousand words. By building a L2S model with this seed lexicon, the authors were able to generate candidate pronunciations for a 30,000 word lexicon. These candidates were then pruned according to how well they fit the acoustic training data, resulting in a lexicon nearly as good as an expert-defined phonetic lexicon. McGraw et al. [2] assumed the existence of an expert-created phonetic pronunciation lexicon which was used to train a letter to sound (L2S) model. This model was then used to generate a large number of new candidate pronunciations for each word in the recognizer's vocabulary, and a Pronunciation Mixture Model (PMM) was then applied to assign weights to those candidate pronunciations. Similar approaches to these were performed by [3, 4].

The work most closely related to ours was performed by Hartmann et al. [5] and Lee et al. [6], neither of whom utilized an expert seed lexicon but rather assumed that utterance-level orthographic transcriptions were available. The framework presented by Hartmann et al. first constructed a grapheme-based recognizer, and then clustered the context-dependent grapheme acoustic models into a new set of units. A phrase-based machine translation model was applied to adapt the initial lexicon to better fit the new acoustic units. The authors reported a 13% relative improvement in word error rate (WER) over the grapheme baseline on the Wall Street Journal corpus. Lee et al. proposed a Bayesian graphical model mapping orthographic transcriptions to acoustics through several layers of latent hierarchy. Inference within the model allowed the authors to recover a set of acoustic models, and a pronunciation lexicon. The authors reported a 15% relative reduction in WER over a grapheme baseline for an English weather query recognition task.

1.2. Motivation and Contributions

Many languages utilize orthographies that are phonetic to a varying degree [7, 8]. Context-dependent systems model trigraphemes much in the same vein that phone-based recognizers model triphones, typically improving upon context-independent grapheme-based systems. While context-dependent modeling no doubt benefits from its ability to model co-articulation of sounds, it can also compensate to a certain degree for the systematic errors introduced by the invalid assumption that every letter maps to its own unique phoneme. However, this compensation is not perfect, as grapheme-based recognizers are often more error prone than their phone-based counterparts. The freedom to do away with a pronunciation lexicon altogether gives grapheme-based systems appeal and reduces the cost of deploying recognizers to new languages or domains.

This paper presents a data-driven method for constructing a weighted pronunciation lexicon given some initialization, such as a grapheme-based lexicon. We adopt a similar strategy to [2], but with a twist - we do not assume the existence of a phonetic lexicon which can be used to fit a L2S model. Our method exploits a novel technique for hybridizing forced alignment and acoustic unit decoding for the purposes of training the L2S model, given only utterance-level orthographic transcriptions. The L2S model is then used to generate a set of candidate pronunciations for each word seen in the training data, and the PMM is applied to weight each pronunciation. Our experiments

*Supported in part by the Intelligence Advanced Research Projects Activity(IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government

show substantial improvement over the grapheme baseline on our English language task, closing 87% of the performance gap between GMM-based triphone and trigrapheme systems, and 57% of the gap between a more advanced pair of systems. We encounter more difficulty with two IARPA-Babel languages, and provide some simple analysis as to why this might be the case by introducing the *LLG error rate*, a novel measure of joint confusion between the lexicon and language model.

2. Method Overview

2.1. Semi-Constrained Forced Alignment

In order to train the L2S model, we require training data in the form of parallel sequences of graphemes and acoustic units. Often in practice a phonetic pronunciation lexicon is used to fit the L2S model, but in the case of a grapheme-based recognizer the pronunciation lexicon simply maps each grapheme to itself. Hence, a L2S model trained on such data would be trivial and not very useful for the purpose of generating novel pronunciations. We attempt to create a richer L2S model utilizing training data derived from a semi-constrained acoustic alignment technique. We create an utterance-dependent decoding graph for each training utterance which is similar to a forced alignment graph, but replaces every other word with a grapheme-star finite-state transducer (FST). The motivation behind the design of this FST is that it anchors the acoustics to the transcription, while still allowing the decoding path to explore alternative acoustic unit sequences for some of the words in the utterance.

More explicitly, let $W = w_1, w_2, \ldots w_n$ be the words comprising the transcription of utterance U. Also let $L(w)$ represent the FST which only accepts the string of graphemes spelling word w and writes its input symbols to its output. Assuming H and C are the standard HMM state transition and context-mapping FSTs defined by Kaldi [11], the forced alignment FST for U can be written as $H \circ C \circ F$, where $F = L(w_1)L(w_2)\ldots L(w_n)$. Now, let $P = S \circ G$, where S accepts any sequence of graphemes and writes its input symbols to its output, G is a language model over graphemes (a simple bigram model trained on the graphemic lexicon), and $\circ$ denotes the FST composition operator. To perform semi-constrained forced alignment on utterance U, we decode with the graph $H \circ C \circ \hat{F}$, where $\hat{F} = PL(w_2)PL(w_4)PL(w_6)\ldots PL(w_n)$, assuming in this case that n is even.

Given a collection of utterances and their transcriptions, we construct the $\hat{F}$ graph for each individual utterance and then decode the acoustics to obtain a 1-best grapheme sequence. We take each utterance's decoded grapheme sequence and build a pseudo-lexicon of the form $g_1, \ldots, g_m : s_1, \ldots, s_n$, where $g_1, \ldots, g_m$ denotes an utterance's transcription and $s_1, \ldots, s_n$ the 1-best acoustic unit sequence. This pseudo-lexicon is then used to train our L2S model. It should be noted that we produce the acoustic unit sequences in terms of context-independent labels, even though context-dependent acoustic models are used for decoding. Other approaches [5] took care to avoid this extra source of constraint, but our experiments on English demonstrate improved performance despite this constraint.

2.2. Letter to Sound Modeling

One way to view the problem of constructing a lexicon is fitting a distribution $P(b|w)$ over pronunciations b for each word w in the vocabulary. In order to tractably estimate $P(b|w)$ for each word in the vocabulary, we specify a finite support for each word-specific distribution over pronunciations. To generate these candidate pronunciations, we use a letter-to-sound (L2S) model which can probabilistically map sequences of graphemes to sequences of sound units. Specifically, we use the Bisani-Ney joint sequence model [9], although any L2S model capable of producing multiple pronunciations for a given word could be applied. The full details of the Bisani-Ney model can be found in [9], so we provide only a brief review of the essentials.

To model P(w, b), [9] employed a model whose fundamental unit, the graphone, was assumed to jointly generate letters and sounds. Here, we restrict our attention to the singular case, in which a graphone may map at most 1 letter to at most 1 sound. We also use the Bisani-Ney model to map between graphemes representing letters and graphemes representing sound units; however, for consistency we will use the word "graphone". A singular graphone takes the form $g = letter : sound$, although we allow the mappings $letter : \epsilon$ and $\epsilon : sound$ to handle insertions and deletions. A sequence of graphones uniquely specifies a sequence of letters and sounds, but a joint sequence of letters and sounds may be associated with many graphone sequences. For this reason, estimating an n-gram language model over graphone units given parallel letter/sound sequences requires the use of an EM algorithm, for which [9] provides an open source implementation. Once an n-gram model over graphones is trained, it can be represented as a weighted FST that reads letter units and writes sound units. Given a sequence of letters w, the N-best paths through the FST predict the N most probable pronunciations for w.

2.3. Pronunciation Mixture Modeling

Speech recognition is typically motivated by a "fundamental" equation inspired by Bayesian signal recovery across a noisy channel. Assuming an acoustic signal A was observed, the goal of speech recognition is posed as finding the most likely sequence of words $W^* = w_1^*, \ldots, w_n^*$ which gave rise to A. Via Bayes' Rule, this is often stated mathematically as

$$W^* = \arg\max_W P(A|W)P(W). \quad (1)$$

$P(A|W)$ is often referred to as the acoustic model, and $P(W)$ the language model. HMM-based recognizers often assume canonical pronunciations for the words in the recognizer's lexicon, which specify the sequence of sub-word acoustic models that are used to build an HMM for each word in the vocabulary. To explicitly introduce stochasticity across a set of multiple pronunciations that may be used for each word, we let $\mathcal{B}$ represent all possible sequences of sub-word units. An utterance's pronunciation sequence is then defined as $B = b_1 b_2 \ldots b_n, b_i \in \mathcal{B}$. Assuming that a word's acoustic realization is independent of the word given its pronunciation, Eq. 1 then becomes

$$W^* = \arg\max_W P(W) \sum_{B \in \mathcal{B}} P(A|B)P(B|W). \quad (2)$$

In practice, we use the Viterbi approximation and replace the summation over all pronunciations with a maximization:

$$W^* = \arg\max_W \max_B P(A|B)P(B|W)P(W) \quad (3)$$

Eq. 3 expresses how pronunciation probabilities can be explicitly included in the speech recognition search problem, although we must still consider how to parameterize and estimate $P(B|W)$, which is where the Pronunciation Mixture Model comes into play. Although [2] provides a more complete theoretical treatment of the PMM, we present here the

technique used in our experiments. First, given any utterance, we assume that each word's pronunciation is independent of the surrounding pronunciations, $P(B|W) = P(b_1|w_1)P(b_2|w_2)\dots P(b_n|w_n)$. Further, we assume that for each word w, $P(b|w)$ is a categorical distribution with a finite support which has been predetermined by computing the N-best pronunciations for w according to the L2S model. We define the PMM model parameters $\theta_{b|w} = P(b|w)$ and initialize each to be proportional to the L2S model's score for pronunciation b given w. We then form a lexicon L which includes all candidate pronunciations for all words in the vocabulary. Given existing acoustic and language models, we use this new lexicon to perform forced-alignment of the training acoustics constrained by their transcriptions. For each utterance, we reserve the N-best list of phonetic paths from this alignment while also retaining word boundaries. Normalizing the log-likelihoods across each N-best list yields an approximation to the posterior probability $P(B|u, W; \theta)$ where u is the acoustic observation and W the corresponding word transcription. This posterior can be used to re-estimate the PMM model parameters θ. Given a training set (which may be the same set used to train the acoustic and language models) of M utterances $D = \{u_i, W_i\}$, where u_i is the acoustic observation for utterance i, and W_i is the corresponding word transcription, the EM update equations for ML estimation of θ taken from [2] are:

$$\bar{\mathrm{M}}_\theta[w, p] = \sum_{i=1}^{M} \sum_{B \in \mathcal{B}_i} P(B|u_i, W_i; \theta) \cdot M[p, w, W_i, B] \quad (4)$$

$$\theta^*_{p|w} = \frac{\bar{\mathrm{M}}_\theta[w, p]}{\sum_{p' \in \mathcal{B}} \bar{\mathrm{M}}_\theta[w, p']} \quad (5)$$

where $\mathcal{B}_i$ is the set of unique pronunciation paths appearing in the N-best list for the ith utterance, and $\mathrm{M}(p, w, W_i, B)$ is the number of times word w was aligned with pronunciation p in the N-best list for the i^{th} utterance. Note that we do not re-decode the data between subsequent EM iterations, but rather adjust the posteriors in each N-best list according to the updated θ.

3. Experiments

3.1. Corpora

Our English language experiments were performed on the Jupiter corpus [10]. This corpus is comprised of telephone queries to an automated system providing weather information. The queries are relatively short, on average consisting of 6 words, and covering a vocabulary of 1,805 unique words. We use an 80 hour set of queries to train our acoustic and language models, as well as the relearned lexicon. For testing, we use a 3,497 utterance test set, consisting of 3.18 hours of speech. For our Haitian and Lao experiments, we utilized data from the IARPA Babel Project. Both systems were trained on the LimitedLP data, from respective releases IARPA-babel203b-v2.1a and IARPA-babel201b-v0.2b. These training sets each contain approximately 10 hours of conversational telephone speech. The vocabulary sizes for each of these languages are considerably larger than the Jupiter vocabulary, with 6,361 unique words appearing in the Lao training data, and 4,838 unique words in the Haitian training set. For testing, we use the 10 hour conversational development sets that accompany the training sets.

3.2. Experimental Conditions

We utilize the Kaldi speech recognition toolkit [11] to construct the HMM-GMM recognizers used in our experiments. Our baseline English system was trained using the standard 39-dimensional MFCC feature representation, with cepstral mean and variance normalization applied on a per-utterance basis. We first flat-start trained 26 context-independent grapheme acoustic models, then used these models to bootstrap the training of a context-dependent tri-grapheme system. The tri-grapheme system was trained with 1300 context-dependent states, with approximately 8 Gaussians per state. In our results, we refer to this system as "GMM". Finally, we stacked the 13-dimensional MFCC feature vector belonging to each frame with its previous three and subsequent three neighboring feature vectors, and then applied LDA and MLLT to the results, bringing the final feature dimension down to 40 [12]. We then retrained the acoustic models using these features, using the "GMM" system for initial bootstrapping. In our results, we refer to this system as "LDA". For decoding, a trigram language model with modified Kneser-Ney smoothing was used.

The Haitian and Lao systems were trained using PLP features with the addition of a pitch estimate and probability of voicing as two extra features [13]. Conversation-level mean and variance normalization was applied to these features, which were then stacked with delta and double delta features to train context-independent grapheme acoustic models, which were then used to bootstrap the training of context-dependent acoustic models. Finally, the 15-dimensional frame feature vectors were stacked with their previous 4 and subsequent 4 frames before applying LDA and MLLT to reduce their dimensionality down to 40. The acoustic models were retrained on these features to yield the system referred to as "LDA" in our experiments. For both languages, we used 4-gram Good-Turing smoothed language models during decoding.

We applied our lexicon relearning method on top of the already retrained recognizers, referred to as "RL" in our results. After performing semi-constrained acoustic unit decoding, we trained a unigram graphone model which was used to generate approximately 30 candidate pronunciations per word. We then applied the PMM, using $N = 10$ for the N-best list size. Lastly, we renormalized the pronunciation weights such that the highest scoring pronunciation for each word received a weight of 1.0, and then threw away any pronunciation with a weight less than 0.1. We chose this method over normalizing each word's set of pronunciations to a probability distribution because the latter penalizes words with more pronunciations (under the Viterbi approximation used in decoding), and we have found the former to work better in practice. After a lexicon has been relearned, we can use it to re-align the training data with the new pronunciations and then retrain the acoustic model set, referred to as "RAM" in our results.

3.3. Results and Discussion

Our experimental results are enumerated in Table 1. Our lexicon re-learning method provided significant improvements for the English language systems, closing 86% of the gap between the triphone GMM system and the tri-grapheme GMM system. The extra discriminative power afforded by the LDA features shrunk the WER gap between the triphone and tri-grapheme systems to only 0.53% absolute, but relearning the lexicon and acoustic models was still able to bridge 57% of this gap. For the curious, a few example utterances from the semi-constrained acoustic unit decoding step are displayed in Table 2, and a few example pronunciations learned are shown in Table 3.

Both the Lao and Haitian recognizers began with very high WERs. Relearning the Haitian lexicon provided a small 0.2%

Lang	System	WER (%)
Eng.	Gr. GMM	10.6
Eng.	Gr. GMM + RL	9.9
Eng.	Gr. GMM + RL + RAM	9.4
Eng.	Ph. GMM	9.2
Eng.	Gr. LDA	8.5
Eng.	Gr. LDA + RL	8.4
Eng.	Gr. LDA + RL + RAM	8.2
Eng.	Ph. LDA	8.0
Lao	Gr. LDA	69.9
Lao	Gr. LDA + RL	72.6
Lao	Gr. LDA + RL + RAM	71.5
Hait.	Gr. LDA	73.6
Hait.	Gr. LDA + RL	73.4
Hait.	Gr. LDA + RL + RAM	73.9

Table 1: Results across our three languages. "Ph." denotes a phonetic system, and "Gr." denotes a graphemic system.

transcript	sound unit sequence
hellojupiter	**w o u l o** j u p i t e r
nicefrance	**s n e a s e** f r a n c e
rhodeisland	**r o u d a** i s l a n d
iwantedsouthtexas	**w o w** w a n t e d **s o u t** t e x a s

Table 2: Examples the semi-constrained acoustic unit decoding on Jupiter. Bold graphemes correspond to the parts of the path that went through a grapheme-star FST.

absolute improvement to WER, but after retraining the acoustic models the system's performance became worse than the baseline. For Lao, the relearned lexicon was a detriment to WER before and after relearning the acoustic models, getting several percentage points worse. We performed some simple analysis as to why our lexicon learning scheme fared much better on English versus the Babel languages. To this end, we defined a simple metric which we call the *LLG error rate*. To compute this, we first construct the FST $S = inv(L) \circ L \circ G$, where L is a lexicon FST mapping sound unit sequences to words, G is a language model, and $inv(L)$ represents the inverse FST of L which maps words to sound unit sequences. Given an utterance whose transcript contains the word sequence W, we construct the FST T which simply writes W to its output tape. The 1-best path through $T \circ S$ yields an output sequence of words $\hat{W}$, with which we compute a standard word error rate against the reference, W. We apply this to each test set, omitting any utterances with out-of-vocabulary words, and accumulate all of the errors to form a final LLG error rate. The intuition behind the LLG error rate is that it provides an approximation as to how confusable the pronunciations in the lexicon are with one another under the constraints of the language model, without directly taking into account an acoustic model or any acoustic data.

Table 4 shows that both Babel languages begin with significantly higher LLG error rates than English, and that the expansion of the lexicon introduces enough new pronunciations to significantly increase the LLG error rate. The table also shows that the language model perplexities of both Babel languages' test sets were far higher than that of the English set. It should be noted that the English system was trained on approximately 8 times as much data as either of the Babel languages, and the speech was not conversational in nature as was the case with the Babel languages. That said, our results suggest that expanding the lexicon can create significantly more ambiguity for the recognizer under very weak language model constraints.

word	weight	pronunciation
switzerland	1.0	s w i t s e l a n d
switzerland	0.88	s w i t s e r l a n d
switzerland	0.34	s w i t z e l a n d
switzerland	0.19	s w i t s e r l a n
prediction	1.0	p r d i c t i o n
prediction	0.31	p r e d i c t i o n
toronto	1.0	t o r o n t o
toronto	0.16	t r o n t o
toronto	0.14	t o r o n o
toronto	0.14	t o r o n t a

Table 3: Example pronunciations learned for Jupiter.

Lang	LLG	LLG-R	PPW	PPL
Eng.	0.07	1.63	3.29	8.43
Lao	1.54	8.36	1.26	167.16
Hait.	7.46	31.04	1.46	146.35

Table 4: LLG error rates for the grapheme lexicons (LLG) and relearned lexicons (LLG-R). Also shown are the average number of pronunciations per word (PPW) for the relearned lexicons, and language model perplexities on the test sets (PPL)

4. Conclusion

We have presented a method for automatically learning an expanded pronunciation lexicon given an initial grapheme recognizer, without assuming the existence of any expert pronunciations. We have demonstrated significant improvement on English, but encountered more difficulty when faced with resource-limited languages. However, it is possible that the relearned lexicons could provide some benefit for a spoken term detection task. We have also introduced the LLG error rate, a measure of joint confusion inherent in a lexicon and language model. While this measure does not appear to be directly correlated with word error rate, it seems to provide some insight as to how well a language model can overcome pronunciation ambiguity. It is possible that the LLG error rate could be used as an analytic tool for determining what kinds of errors can be attributed to the acoustic models or to the lexicon and language models, a topic which we believe merits further investigation.

In a resource-constrained scenario, discriminative training of the PMM may also prove to be useful. It is also possible that incorporating a more explicit prior distribution over pronunciations into the PMM estimation would prevent spurious pronunciations from being learned for words with a limited number of training examples; a prior which placed additional weight on the graphemic pronunciation of each word would be a good place to start. Although we have applied our method to graphemic systems in this paper, it would be possible to relearn any kind of lexicon in the manner we describe here, be it phonetic or perhaps in terms of a set of unsupervised units. We leave this, as well as the aforementioned investigations, to future work.

5. Acknowledgements

The authors would like to thank Ekapol Chuangsuwanich for his valuable insights.

6. References

[1] L. Lu, A. Ghoshal, and S. Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, 2013.

[2] I. McGraw, I. Badr, and J. Glass, "Learning lexicons from speech using a pronunciation mixture model," in *IEEE Transactions on Audio, Speech, and Language Processing*, February 2013, Volume 21, Issue 2, pp. 357-366.

[3] N. Goel, S. Thomas, M. Agarwal, P. Akyazi, L. Burget, K. Feng, A. Ghoshal, O. Glembek, M. Karafiat, D. Povey, A. Rastrow, R. Rose, and P. Schwarz, "Approaches to automatic lexicon learning with limited training examples," in *Proc. of IEEE ICASSP*, 2010.

[4] R. Rasipuram and M. Magimai-Doss, "Combining acoustic data driven G2P and letter-to-sound rules for under-resourced lexicon generation," in *Proc. of Interspeech,* 2012.

[5] W. Hartmann, A. Roy, L. Lamel, and J.L. Gauvain, "Acoustic unit discovery and pronunciation generation from a grapheme-based lexicon," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, 2013.

[6] C.Y. Lee, Y. Zhang, and J. Glass, "Joint learning of phonetic units and word pronunciations for ASR," in Proc. of Empirical Methods in Natural Language Processing (EMNLP), Seattle, Washington, USA, 2013.

[7] M. Killer, "Grapheme-based speech recognition," M.S. thesis, Carnegie Mellon University, 2003.

[8] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large-vocabulary speech recognition," in *Proc. of IEEE ICASSP*, 2002.

[9] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," in *Speech Communication*, May 2008, Volume 50, Issue 5, pp. 434-451.

[10] 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," in *IEEE Transactions of Speech and Audio Processing,* vol. 8, no. 1, pp. 85-96, Jan. 2000

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[12] M. J. F. Gales, "Maximum likelihood linear transformation for HMM-based speech recognition," in Computer Speech and Language, 1998.

[13] P. Ghahremani, B. Baba Ali, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," to appear in *Proc. of the IEEE Internation Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 2014.