ACOUSTIC DATA-DRIVEN PRONUNCIATION LEXICON GENERATION FOR LOGOGRAPHIC LANGUAGES

Guoguo Chen[1], Daniel Povey[1,2], Sanjeev Khudanpur[1,2]

[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
guoguo@jhu.edu, dpovey@gmail.com, khudanpur@jhu.edu

ABSTRACT

Handcrafted pronunciation lexicons are widely used in modern speech recognition systems. Designing a pronunciation lexicon, however, requires tremendous amount of expert knowledge and effort, which is not practical when applying speech recognition techniques to low resource languages. In this paper, we are interested in developing speech recognition systems for logographic languages with only a small expert pronunciation lexicon. An iterative framework is proposed to generate and refine the phonetic transcripts of the training data, which will then be aligned to their word-level transcripts for grapheme-to-phoneme (G2P) model training. The G2P model trained this way covers graphemes that appear in the training transcripts (most of which are usually unseen in a small expert lexicon for logographic languages), therefore is able to generate pronunciations for all the words in the transcripts. The proposed lexicon generation procedure is evaluated on Cantonese speech recognition and keyword search tasks. Experiments show that starting from an expert lexicon of only $1K$ words, we are able to generate a lexicon that works reasonably well when compared with an expert-crafted lexicon of $5K$ words.

Index Terms— Pronunciation lexicon, logographic language, speech recognition, keyword search

1. INTRODUCTION

In the past few years there has been an increased interest in developing speech recognition and keyword search systems for low resource languages. Building a speech recognition system for a new language usually requires three major resources: first, transcribed speech data for acoustic modeling; second, optional additional text data for language modeling; and finally a lexicon that maps words to sub-word modeling units, typically, phonemes. While it is relatively easy to collect transcribed speech data and text data, the creation of the pronunciation lexicon is often expensive as it requires large amount of expert knowledge and effort. The pronunciation lexicon, therefore, is the Achilles heel when building speech recognition systems for low resource languages.

A lot of techniques have been proposed in the literature to reduce the expert effort needed in lexicon design for automatic speech recognition. One solution is to model graphemes instead of phonemes as the sub-word units, which completely removes the necessity of a phonetic pronunciation lexicon in speech recognition. Such techniques have found success in languages with alphabetic (a.k.a. segmental) writing systems [1, 2], but cannot naturally be extended to other writing systems, e.g., logographic, as the graphemes in those languages do not necessarily imply the phonetic representation of the words, and the number of graphemes is often quite large, e.g., few thousands. Other researchers have been looking into techniques that generate pronunciation lexicons in a data-driven and stochastic manner. In [3, 4], a hierarchical Bayesian model is proposed to jointly discover the phonetic inventory as well as the grapheme-to-phoneme (G2P) mapping rules using only transcribed speech data. The authors show encouraging results in their paper, but the pronunciation lexicon discovery process itself is quite time consuming with the proposed model, making it the bottleneck when rapid development of speech recognition systems is desired. A more practical technique is to start from a small expert pronunciation lexicon, enlarge it by learning the pronunciations of additional words and incorporate them into the existing speech recognition system. In [5, 6, 7, 8, 9, 10], the expert lexicon is used to train a G2P model, with which pronunciations of additional words are generated, and added to the existing lexicon. The enlarged lexicon can then be refined in a data-driven manner.

We are interested in developing speech recognition systems for logographic languages with only a small expert pronunciation lexicon. We follow the general techniques in [5, 6, 7, 8, 9, 10], where G2P conversion is used to generate pronunciations for out-of-vocabulary (OOV) words. For logographic languages, due to the large number of unique graphemes, G2P models trained on a small seed lexicon, as proposed in [9], typically are not able to generate pronunciations for all the words in the training transcripts. Previous work on pronunciation modeling for logographic languages such as Mandarin Chinese mostly only focus on pronunciation variants [11, 12], and does not address the problem of unseen graphemes. In this paper, we propose to incorporate the phonetic transcripts of the training data into G2P modeling through an iterative framework, so that all the graphemes that appear in the training transcripts will be modeled. We start from the initial expert lexicon and build a bootstrap speech recognition system, with which we generate phonetic transcripts for the training data. These phonetic transcripts are aligned to their word-level transcripts using a many-to-many alignment algorithm [13], which can then be used for G2P modeling and lexicon update. This procedure is carried out iteratively, and is able to generate pronunciations for words in the training transcripts.

The remainder of this paper is organized as follows. We describe our pronunciation generation method in Section 2, and explain how

This work was partially supported by NSF Grant No 1005411 and 0963898, and DARPA Contract No HR0011-15-2-0027. We also thank Eleanor Chodroff from Johns Hopkins University for the meaningful discussion on writing systems, and Ekapol Chuangsuwanich from Massachusetts Institute of Technology for his valuable suggestions.

we handle multiple pronunciations in our speech recognition system in Section 3. We then illustrate our proposed iterative lexicon generation framework in Section 4. The experimental setup is detailed in Section 5, and results are provided in Section 6. Finally we reiterate our main claims in Section 7.

2. PRONUNCIATION GENERATION FOR LOGOGRAPHIC LANGUAGES

Grapheme-to-phoneme (G2P) conversion is a task to map a word, represented by a sequence of graphemes, to its pronunciation, represented by a sequence of phonemes [14]. Suppose w is the grapheme sequence of a word, and $\hat{p}$ its corresponding pronunciation, G2P can be framed as follows [15]:

$$\hat{p} = \arg\max_{p} P(p|w) = \arg\max_{p} P(w, p) \tag{1}$$

Therefore, one can either model the conditional distribution $P(p|w)$ or the joint distribution $P(w, p)$. In this paper, we choose to model the later.

The G2P formulation generally works for both alphabetic and logographic languages. For logographic languages, since there are usually large number of unique graphemes, special care should be taken during model training. Below we explain how we generate pronunciations for logographic languages when we build a speech recognition system with a small expert lexicon.

2.1. G2P training data

G2P training data usually consists of a set of grapheme sequences and their corresponding phoneme sequences. For G2P conversion in speech recognition tasks, oftentimes the pronunciation lexicon is used for G2P training.

In our task, since we are building speech recognition systems for logographic languages and we only have a small expert lexicon, training the G2P model using the lexicon will result in a large number of unseen graphemes in the training data, and the G2P model trained this way will not be able to generate pronunciations for all the words in the training data.

In order to cover as many graphemes as possible, we propose to train the G2P model on the phonetic training transcripts, in addition to the pronunciation lexicon. A speech recognition system can first be trained with the small expert lexicon that we start with. Training data can then be decoded into phone sequences using this initial speech recognizer. Now for each utterance in the training data, we have a sequence of graphemes, and their corresponding phonemes. We use those grapheme and phoneme sequences together with the pronunciation lexicon to train the G2P model, which will be able to generate pronunciations for all the words in the training transcripts. The phone sequences generated by the initial recognizer will generally be noisy, but it can be refined through an iterative procedure, which we will explain in details in Section 4.

2.2. G2P alignment

Grapheme and phoneme sequences provided in the G2P training data have to be aligned into modeling units called "graphones" before building the pronunciation model. For logographic languages such as Cantonese, one grapheme typically corresponds to multiple phonemes, we adopt a many-to-many alignment algorithm proposed in [13]. We use the open source toolkit Phonetisaurus [16] for our alignment, which implements a weighted finite state transducer (WFST) version of the many-to-many aligner. In our Cantonese experiments, we allow at most 1 grapheme, and at most 4 phonemes in a single graphone. A typical Cantonese graphone generated by Phonetisaurus looks like " 中 / dz1 u:1 N1", where " 中" is a single grapheme (Chinese character), and "dz1 u:1 N1" is the phoneme sequence aligned to it.

2.3. Joint n-gram model

After aligning graphemes and phonemes into graphones, a 4-gram language model is trained for the graphone sequences using SRILM [17]. The language model is further converted into a WFST G, whose input labels are graphemes, output labels are phonemes, and weights are the n-gram scores. The WFST G serves as the pronunciation model.

2.4. Pronunciation generation

Given a grapheme sequence w of a certain word, generating pronunciation p for the word is essentially finding the best path through G that has input label sequence w, as described in Equation 2.

$$p = \mathrm{ShortestPath}(\mathrm{Determinize}(\mathrm{Project}(w \circ G))) \tag{2}$$

For our Cantonese experiments, we generate at most 5 different pronunciations for a single word.

3. HANDLING MULTIPLE PRONUNCIATIONS

Since multiple pronunciations are generated for words that are not already in the small expert lexicon, we explicitly model pronunciation and inter-word silence probability as that has been found useful when pronunciation variants exist in the lexicon [18]. Unlikely pronunciations are further pruned away based on the estimated pronunciation probabilities. We incorporate the estimated pronunciation and inter-word silence probabilities into the lexicon transducer, which will be used in both training and decoding.

3.1. Pronunciation probability estimation

We estimate the pronunciation probabilities for a word with multiple pronunciations via simple relative frequency [19, 20, 21]. Let $w.p_i$ be the i^{th} pronunciation of word w, $1 \leq i \leq N_w$, and N_w is the number of different *baseform* pronunciations of word w in the lexicon. Let $C(w, w.p_i)$ be the count of "$w\ w.p_i$" pairs in the aligned training data. The probability of a pronunciation $w.p_i$ given the word w is simply

$$\pi(w.p_i|w) = \frac{C(w, w.p_i) + \lambda_1}{\sum_{i=1}^{N_w}(C(w, w.p_i) + \lambda_1)}, \tag{3}$$

where λ_1 is a smoothing constant that we typically set to 1. An undesirable consequence of (3) is that a word with several equiprobable pronunciations is unfairly handicapped w.r.t words that have a single pronunciation. Max-normalization, whereby the pronunciation probabilities are scaled so that the most likely pronunciation of each word has "probability" 1, has been found helpful in speech recognition [22]. This suggests using

$$\pi'(w.p_i|w) = \frac{\pi(w.p_i|w)}{\max_{1 \leq i \leq N_w} \pi(w.p_i|w)}. \tag{4}$$

We do max-normalization for pronunciation probabilities in all our experiments. The quantity $\pi'(w.p_i|w)$ is of course not a well defined probability any more.

3.2. Silence probability estimation

This section explains how we model the probability of silence preceding or following certain pronunciation. For a given sequence of words, we assume there is either a silence or non-silence event between two consecutive words. Since such an event usually depends on the neighbouring words, we further assume that it only depends on the two surrounding words, i.e., we model the event using $P(s \,|\, w.p_i, w'.p_j)$ and $P(n \,|\, w.p_i, w'.p_j)$, where $w.p_i$ and $w'.p_j$ are the surrounding pronunciations, s and n represent silence and non-silence event. For computation simplicity, we decompose this into two parts: (i) probability of inter-word silence (or non-silence) following the pronunciation, and (ii) probability of inter-word silence (or non-silence) preceding the pronunciation.

We use $P(s_r|w.p)$ to denote the probability of inter-word silence following the pronunciation $w.p$, and $P(n_r|w.p)$ for the complementary probability of non-silence following $w.p$. We compute $P(s_r|w.p)$ from training data counts as

$$P(s_r|w.p) = \frac{C(w.p\, s) + \lambda_2 P(s)}{C(w.p) + \lambda_2}, \tag{5}$$

where $C(w.p\, s)$ is the count of the sequence "$w.p\, s$" in the training data alignment, $C(w.p)$ is the count of pronunciation $w.p$, $P(s) = C(s)/(C(s)+C(n))$ is the overall probability of inter-word silence, and λ_2 is a smoothing constant that we set to 2 for experiments.

Directly modeling the probability of inter-word silence (or non-silence) preceding the pronunciation will cause double counting problem. We therefore only compute it as correction term instead

$$F(s_l|w'.p) = \frac{C(s\ w'.p) + \lambda_3}{\widetilde{C}(s\ w'.p) + \lambda_3}, \text{ and} \tag{6}$$

$$F(n_l|w'.p) = \frac{C(n\ w'.p) + \lambda_3}{\widetilde{C}(n\ w'.p) + \lambda_3}, \tag{7}$$

where $\widetilde{C}(s\ w'.p)$ and $\widetilde{C}(n\ w'.p)$ are the "mean" counts of silence or non-silence preceding $w'.p$, estimated according to $\widetilde{C}(s\ w'.p) = \sum_v C(v\ *\ w'.p) P(s_r|v)$, where the sum is over all pronunciations v in the lexicon, the symbol "$*$" in $C(v\ *\ w'.p)$ denotes either s or n, and $P(s_r|v)$ is computed using Equation (5). λ_3 is a smoothing constant that we set to 2 for experiments reported here.

Putting it all together, we estimate the inter-word silence (or lack thereof) given the neighbouring words as follows

$$\begin{aligned} P(s \,|\, w.p_i, w'.p_j) &\approx P(s_r \,|\, w.p_i) \times F(s_l \,|\, w'.p_j), \text{ and} \\ P(n \,|\, w.p_i, w'.p_j) &\approx P(n_r \,|\, w.p_i) \times F(n_l \,|\, w'.p_j). \end{aligned}$$

3.3. Pronunciation selection

Pruning of pronunciations is performed after estimating the pronunciation and silence probabilities. For each word, we only keep the pronunciations with probability higher than 0.6. Note that this is the "max-normalized" probability, therefore each word may have multiple pronunciations after pruning.

4. ITERATIVE FRAMEWORK

Our pronunciation generation procedure follows a general iterative learning schedule. Unlike [9] and [7], where the iterative procedure is more focused on selecting the best pronunciation given the pronunciations generated by the G2P model, our framework attempts to refine the phonetic transcripts generated by the speech recognizer, which will be used to train the G2P model together with the small expert lexicon, as described in Section 2.1.

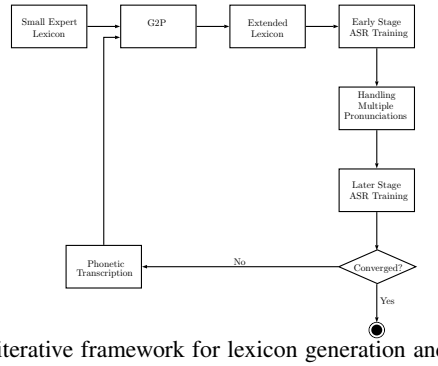

Fig. 1. An iterative framework for lexicon generation and acoustic modeling

Figure 1 illustrates our proposed iterative framework for lexicon generation and acoustic modeling. We start from a small expert pronunciation lexicon of 500 or 1000 words. Pronunciations of the words that are already in the small expert lexicon are kept untouched throughout the whole process. The small expert lexicon is used to train an initial G2P model, which is then applied to the words in the training transcripts to create an extended lexicon. We generate at most 5 pronunciations for each word in the training transcripts, if it is not already in the expert lexicon. Note that the initial extended lexicon typically cannot cover all the words in the training transcripts, due to the large number of unseen graphemes that are not covered by the small expert lexicon. Those words are treated as OOV words in the first iteration. For later iterations, since we add phonetic transcripts for G2P training, the extended lexicon will be able to cover all the words in the training transcripts.

Early stages of acoustic model training are carried out with this extended lexicon, typically till the speaker adapted training stage. Since the extended lexicon contains multiple pronunciations for each word, we estimate the pronunciation and inter-word silence probabilities as described in Section 3, which has found its success when pronunciation variants exist in the lexicon [18]. We further prune away pronunciations with low (less than 0.6) "max-normalized" probabilities. The updated lexicon, as well as the pronunciation and inter-word silence probabilities are incorporated into a new lexicon transducer for later stages of acoustic modeling as well as decoding.

After training the speech recognition system, if the word error rate (WER) performance on some held-out dataset converges to a stable point, we stop the process. Otherwise, we use the trained system to decode the training data into phone sequences, which will be sent back to re-train the G2P system, and start the process again, till convergence. Note that in our experiments we set the language model weight to 0 during the phonetic decoding process, but it may also make sense to use a n-gram phone language model.

5. EXPERIMENTAL SETUP

5.1. Corpus

We evaluate the proposed framework using IARPA Babel language Cantonese [1]. The 10 hour subset of the full language pack is used to conduct our experiments. The language pack also comes with an expert lexicon that contains $5.9K$ lexical entries and $5K$ unique words, which covers all the words in the training transcripts. The number of unique characters covered by this lexicon is $2K$. To simulate the situation where only a small expert lexicon is available for

[1] Language collection release `IARPA-babel101b-v0.4c`.

speech recognition system development, we create two seed lexicons by randomly selecting 500 and 1000 words from the original expert lexicon. These two lexicons have $0.63K$ and $0.95K$ unique characters respectively. For the keyword search task, the development keyword list [2] is used for evaluation.

5.2. System description

We use the open source toolkit Kaldi [23] for all our system development and experiments. Standard PLP analysis is employed to extract 13 dimensional acoustic feature, and a maximum likelihood acoustic training recipe is followed to train speaker adaptive models. This is followed by the modeling of pronunciation and inter-word silence probabilities, which updates the lexicon and prunes away unlikely pronunciations. From this point, two different systems are trained: a hybrid deep neural network (DNN) system and a subspace Gaussian mixture model (SGMM) system with boosted maximum mutual information (BMMI) training. For the keyword search task, lattice indexing is further performed to convert lattice of each utterance into a finite-state acceptor with the posterior score, start-time and end-time for each word encoded as a 3-dimensional weight. An inverted index is then created from these individual acceptors, with paths to accept every possible word sequence in the original lattices. This way, keyword search can be done by composing the keyword acceptor with the inverted index. For details of speech recognition and keyword search systems, readers are referred to [24, 25, 26].

6. RESULTS

We report word error rate (WER) for the speech recognition task, and actual term-weighted value (ATWV) for the keyword search task. Numbers of both metrics are reported in percentage.

6.1. Performance

	$L1$ (500 words)		$L2$ (1000 words)	
	WER	ATWV	WER	ATWV
baseline	75.6	2.79	68.1	9.08
Iteration0	72.9	6.76	65.7	13.81
Iteration1	61.6	13.95	58.9	17.87
Iteration2	61.8	15.19	58.4	18.59
Iteration3	**60.9**	**15.38**	**57.8**	**18.91**
oracle	*54.2*	*23.77*	*54.2*	*23.77*

Table 1. WER and ATWV performance of lexicons from different iterations, SGMM BMMI system

	$L1$ (500 words)		$L2$ (1000 words)	
	WER	ATWV	WER	ATWV
baseline	74.3	3.32	67.7	10.86
Iteration0	71.6	8.62	65.1	14.95
Iteration1	61.8	14.91	58.4	19.59
Iteration2	62.1	16.59	59.1	21.08
Iteration3	**61.4**	**17.64**	**57.2**	**22.16**
oracle	*53.8*	*26.23*	*53.8*	*26.23*

Table 2. WER and ATWV performance of lexicons from different iterations, DNN system

Table 1 and 2 present the WER and ATWV performance of the automatically generated lexicons from various iterations. We evaluate two seed lexicons, one with size 500 ($L1$), and the other with size 1000 ($L2$), as described in Section 5.1. The "baseline" in the two tables corresponds to the speech recognition system trained only with the seed lexicon, and the "oracle" represents the system trained with the full $5K$ words expert lexicon. Iteration0 corresponds to the system trained with the initial extended lexicon, which does not contain all the words from the training transcripts. Starting from Iteration1, phonetic transcripts can be generated by the system, so the G2P model trained on that can generate pronunciations for all the words. The iterative procedure is carried out for three times, excluding Iteration0. Further iterations do not help in our experiments.

Let us start by looking at the ATWV numbers. It is clear from the table that ATWV is increasing through the iterations. This implies that the pronunciation lexicon is indeed improving through the iterative framework. The WER improvements, however, are not as steady as the ATWV improvements, although the best numbers are all achieved at Iteration3. We suspect that this is partly because of decoding noise, and partly because we discard the pronunciations when we update the lexicon with the newly trained G2P model. It might make sense to combine the old and new lexicons, and let the speech recognizer pick the best pronunciation.

Note that there is a performance jump from Iteration0 to Iteration1. This is because the initial extended lexicon trained from the expert lexicon only covers a small portion of words in the training transcripts, while the lexicon in Iteration1 covers all.

It is encouraging to see that, starting from a lexicon of 1000 words, which is just one-fifth of the original lexicon, we are able to achieve a WER of 57.2 and ATWV of 22.16 with our proposed iterative framework. This closes 76% of the WER gap and 74% of the ATWV gap between the baseline and the oracle lexicon.

6.2. Impact of phonetic transcripts quality

	SAT transcripts		DNN transcripts	
	WER	ATWV	WER	ATWV
baseline	67.7	10.86	67.7	10.86
Iteration0	65.1	14.95	65.1	14.95
Iteration1	60.2	18.83	58.4	19.59
Iteration2	59.4	20.46	59.1	21.08
Iteration3	**59.2**	**20.66**	**57.2**	**22.16**
oracle	*53.8*	*26.23*	*53.8*	*26.23*

Table 3. WER and ATWV performance on the DNN system, with lexicons trained on the SAT transcripts or DNN transcripts ($L2$ seed lexicon)

Phonetic transcripts of training data can either be generated by a model from speaker adaptive training (SAT), or by DNN model. The later typically yields better quality. Table 3 gives the performance comparison of lexicons generated by the two models. The table suggests that generating high quality phonetic transcripts is essential in our framework.

7. CONCLUSION

We have presented an iterative framework that is capable of generating pronunciation lexicons for logographic languages. This allows us to rapidly build speech recognition systems with limited expert lexicon for those languages. Experiments on Cantonese suggest that by using a seed lexicon of 1000 words, we are able to achieve reasonably well speech recognition and keyword search performance, when compared with an expert-crafted lexicon of 5000 words.

[2] Keyword list release `babel101b-v0.4c_conv-dev`.

8. REFERENCES

[1] David Harwath and James Glass, "Speech recognition without a lexicon-bridging the gap between graphemic and phonetic systems," in *Proceedings of INTERSPEECH*, 2014.

[2] Mark J.F. Gales, Katherine M. Knill, and Anton Ragni, "Unicode-based graphemic systems for limited resource languages," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[3] Chia-ying Lee, Yu Zhang, and James R Glass, "Joint learning of phonetic units and word pronunciations for asr.," in *Proceedings of EMNLP*, 2013, pp. 182–192.

[4] Chia-ying Lee, Timothy J O'Donnell, and James Glass, "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[5] Françoise Beaufays, Ananth Sankar, Shaun Williams, and Mitch Weintraub, "Learning name pronunciations in automatic speech recognition systems," in *Proceedings of ICTAI*. IEEE, 2003, pp. 233–240.

[6] Xiao Li, Asela Gunawardana, and Alex Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*. IEEE, 2007, pp. 130–135.

[7] Nagendra Goel, Samuel Thomas, Mohit Agarwal, Pinar Akyazi, Lukas Burget, Kai Feng, Arnab Ghoshal, Ondrej Glembek, Martin Karafiát, Daniel Povey, et al., "Approaches to automatic lexicon learning with limited training examples," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 5094–5097.

[8] Ramya Rasipuram and Mathew Magimai Doss, "Combining acoustic data driven g2p and letter-to-sound rules for under resource lexicon generation," in *Proceedings of INTERSPEECH*, 2012, number EPFL-CONF-192596.

[9] Liang Lu, Arnab Ghoshal, and Steve Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*. IEEE, 2013, pp. 374–379.

[10] Ian McGraw, Ibrahim Badr, and James R Glass, "Learning lexicons from speech using a pronunciation mixture model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 2, pp. 357–366, 2013.

[11] Xin Lei, Wen Wang, and Andreas Stolcke, "Data-driven lexicon expansion for mandarin broadcast news and conversation speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 4329–4332.

[12] William Byrne, Veera Venkataramani, Terri Kamm, Thomas Fang Zheng, Zhanjiang Song, Pascale Fung, Yunlei Lui, and Umar Ruhi, "Automatic generation of pronunciation lexicons for mandarin spontaneous speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2001, pp. 569–572.

[13] Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif, "Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion.," in *Proceedings of HLT-NAACL*, 2007, vol. 7, pp. 372–379.

[14] Sittichai Jiampojamarn, *Grapheme-to-phoneme conversion and its application to transliteration*, University of Alberta, 2011.

[15] Stanley F Chen et al., "Conditional and joint models for grapheme-to-phoneme conversion.," in *Proceedings of INTERSPEECH*, 2003.

[16] Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose, "Wfst-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding," in *Proceedings of 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, p. 45.

[17] Andreas Stolcke et al., "Srilm-an extensible language modeling toolkit.," in *Proceedings of INTERSPEECH*, 2002.

[18] Guoguo Chen, Hainan Xu, Minhua Wu, Daniel Povey, and Sanjeev Khudanpur, "Pronunciation and silence probability modeling for ASR," in *Proceedings of Interspeech*, 2015.

[19] Thomas Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, no. 2, pp. 171–188, 2005.

[20] Barbara Peskin, Michael Newman, Don McAllaster, Venkatesh Nagesha, Hywel Richards, Steven Wegmann, Melvyn Hunt, and Larry Gillick, "Improvements in recognition of conversational telephone speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1999, vol. 1, pp. 53–56.

[21] Thomas Hain, PC Woodland, Gunnar Evermann, and Dan Povey, "The CU-HTK march 2000 Hub5e transcription system," in *Proceedings of Speech Transcription Workshop*, 2000, vol. 1.

[22] Eric Fosler, Mitch Weintraub, Steven Wegmann, Yu-Hung Kao, Sanjeev Khudanpur, Charles Galles, and Murat Saraclar, "Automatic learning of word pronunciation from data," in *Proceedings of the International Conference on Spoken Language Processing*, 1996.

[23] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukaš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motliček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society.

[24] Jan Trmal, Guoguo Chen, Dan Povey, Sanjeev Khudanpur, Pegah Ghahremani, Xiaohui Zhang, Vimal Manohar, Chunxi Liu, Aren Jansen, Dietrich Klakow, et al., "A keyword search system using open source software," in *Proceedings of Spoken Language Technology (SLT) Workshop*. IEEE, 2014.

[25] Guoguo Chen, Oguz Yilmaz, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Using proxies for oov keywords in the keyword search task," in *Proceedings of the Automatic Speech Recognition and Understanding (ASRU) Workshop*. IEEE, 2013, pp. 416–421.

[26] Guoguo Chen, Sanjeev Khudanpur, Daniel Povey, Jan Trmal, David Yarowsky, and Oguz Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in low resource languages," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013.