INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Acoustic data-driven lexicon learning based on a greedy pronunciation selection framework

Xiaohui Zhang[1], *Vimal Manohar*[1,2], *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

xiaohui@jhu.edu, vimal.manohar91@gmail.com, danielpovey@gmail.com, khudanpur@jhu.edu

Abstract

Speech recognition systems for irregularly-spelled languages like English normally require hand-written pronunciations. In this paper, we describe a system for automatically obtaining pronunciations of words for which pronunciations are not available, but for which transcribed data exists. Our method integrates information from the letter sequence and from the acoustic evidence. The novel aspect of the problem that we address is the problem of how to prune entries from such a lexicon (since, empirically, lexicons with too many entries do not tend to be good for ASR performance). Experiments on various ASR tasks show that, with the proposed framework, starting with an initial lexicon of several thousand words, we are able to learn a lexicon which performs close to a full expert lexicon in terms of WER performance on test data, and is better than lexicons built using G2P alone or with a pruning criterion based on pronunciation probability.

Index Terms: speech recognition, pronunciation lexicon learning

1. Introduction

In the past few years, there has been an growing interest in investigating acoustic data-driven lexicon learning for continuous speech recognition, i.e. automatically obtaining pronunciations of words for which pronunciations are not available , but for which transcribed acoustic data exists. In order to develop ASR systems under limited lexicon resources, one solution is to adopt a graphemic lexicon [1, 2] or acoustic unit discovery methods [3, 4], which totally eliminate the expert efforts for developing a phonetic pronunciation lexicon. In real applications, however, a more common scenario is that we already have a phonetic inventory, and a small expert lexicon for a specific language. Our work focuses on this case, i.e. given a small expert lexicon, we want to derive pronunciations for Out-of-Vocabulary (OOV) words, for which we know the text form and have acoustic examples.

Given a small expert lexicon, the most straightforward way to generate pronunciation candidates for OOV words is to train a Grapheme-to-Phoneme (G2P) [5] model using the seed lexicon and apply it to these OOV words [6, 7, 8]. But for languages like English, and for proper names and abbreviations, G2P does not always give high quality pronunciations. Pronunciations from phonetic decoding can help to fill this gap. Previous work has combined these with G2P-generated pronunciations [9, 10], or added into G2P training examples [7, 11, 12]. In the work we describe here, we use candidates from both G2P and phonetic decoding.

The aspect of the problem that we focus on is candidate pruning. That is, given a set of pronunciation candidates from G2P and phonetic decoding (and maybe some from a manually created lexicon), which subset should we keep? Keeping all the pronunciations is impractical because it would make decoding slow, and also because too many pronunciations tend to hurt ASR performance, even when pronunciation probabilities are used [13].

Previous work on candidate pruning has relied on estimated pronunciation probabilities to determine which candidates should be cut [11, 6, 8, 7, 12]. The main defect with this is that for words with multiple pronunciations, it tends to give us too many minor pronunciation variants (e.g. reflecting co-articulation effects), which is undesirable for ASR. If we rely on pronunciation probabilities alone it is hard to discard those types of variants while keeping variants that come from different meanings of the word.

The core idea of this paper is a likelihood-based criterion for pronunciation-candidate pruning that naturally keeps candidates that are "far apart".

This paper is organized as follows. We discuss how we generate pronunciation candidates in Section 2; we explain how we collect acoustic evidence from training data in Section 3. We explain our likelihood-based pronunciation selection strategy in Section 4. Experimental results on various ASR tasks are provided in Section 5, and we conclude in Section 6.

2. Collecting pronunciation candidates from multiple sources

In our framework, like [10], we first extend the seed lexicon to include OOV words in the training data, using a G2P model trained on the seed lexicon, and then train an acoustic model (AM) using the G2P-extended lexicon. Then we generate alignments for all training data, based on which we then train a bigram phone language model (LM). Using this phone LM and the AM, we construct a phonetic decoder and use it to generate phonetic transcription of training data. For each individual word token in the transcript, we can align it with a phone sequence using timing information from the alignments and phonetic transcriptions. Then for each specific word w, we can compute the

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128 and IARPA Contract No 2012-12050800010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. 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 DARPA, IARPA, DoD/ARL or the U.S. Government. Thanks Dr. Jack Godfrey, Dr. Alan MaCree, Dr. Mengyang Gu, Chloe Haviland for useful discussions.

http://dx.doi.org/10.21437/Interspeech.2017-588

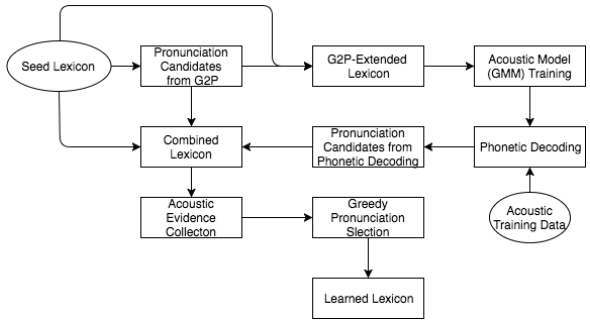

Figure 1: *The proposed framework of acoustic-data driven lexicon learning.*

relative frequency of each phone sequence being aligned to it, by normalizing each phone sequence's count by the most frequent phone sequence's count. Then we filter out those phone sequences whose relative frequency is too low (e.g. smaller than 0.1) and keep the left ones as the alternative pronunciations generated from phonetic decoding. Then we combine these alternative pronunciation candidates with the G2P-extended lexicon into a large lexicon (called combined lexicon). For each word w from the combined lexicon, let $\mathcal{B}$ denote the set of pronunciation candidates collected from multiple sources, and b denote one pronunciation (baseform) candidate. The source of b (denoted as $s(b)$) could be one of the three: G2P/phonetic decoding. In the next section we will specify how we collect acoustic evidence for all pronunciation candidates in $\mathcal{B}$.

3. Acoustic evidence collection

First we introduce some notations. Let $\mathbf{O} = \{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_M\}$ denote acoustic sequences; M_w denote the number of utterances in $\mathbf{O}$ which contain the word w [1]; Then we further define $\theta_{wb} \triangleq p(w, b)$ as the pronunciation probability of a pronunciation b for a word w ($\sum_{b\in\mathcal{B}} \theta_{wb} = 1$), and $\boldsymbol{\theta}_w \triangleq \{\theta_{wb} : b \in \mathcal{B}\}$ as the pronunciation model for word w. We define $\tau_{uwb} \triangleq p(\mathcal{O}_u|w, b)$ as the conditional data likelihood given the pronunciation of w being b, which is determined by the acoustic model. This is the "acoustic evidence" we want to derive from lattice statistics, which is needed by our pronunciation selection algorithm.

With the combined lexicon and an existing AM (the one we used for phonetic decoding in the candidate collection phase), we generate lattices for each training utterance. This lattice generation treats distinct pronunciations of words as distinct symbols for the purposes of lattice determinization, unlike our standard procedure described in [14]. This is achieved by putting both phone symbols and word symbols as the input sequence on the FST prior to lattice determinization. From the lattices, we can obtain per-utterance lattice pronunciation-posterior statistics $\gamma_{uwb} \triangleq p(w, b|\mathcal{O}_u)$.

When the lattices were generated, we assign uniform priors over all pronunciation candidates of each word in the combined lexicon. By Bayes' rule, we can directly use the posterior statistics γ_{uwb} as the likelihoods τ_{uwb} [2]. Because lattices are pruned, a posterior γ_{uwb} could be zero even if w actually appears in a utterance u. So we always floor γ_{uwb} to a small positive scalar δ (In practice it's set between 10^{-7} and 10^{-5}), so that we have $\tau_{uwb} \geq \delta, \forall u, w, b$.

Based on γ_{uwb}, we can obtain another useful statistic, the average pronunciation posterior $\gamma_{wb} \triangleq \frac{1}{M_w}\sum_u \gamma_{uwb}$, where the summation $\sum_u$ is only taken over those utterances where the word w actually appears.

After the lattices were dumped, for each word, we prune away its pronunciations whose average posterior γ_{wb} is too low (e.g. only keeping the top 10), construct a new combined lexicon, and then re-generate the lattices and re-collect acoustic evidence in the same way. We found this pruning is always helpful as it improves the accuracy of the posteriors.

4. Data-likelihood-reduction based greedy pronunciation selection

We formulate the pronunciation selection process as a greedy model selection procedure, with data-likelihood-reduction as the selection criterion. In this section, we'll first specify how to compute the optimal data likelihood given a set of pronunciation candidates using EM and propose a pronunciation selection criterion based on likelihood reduction, and then use an illustrative example to compare the proposed selection criterion against other criteria. At last we talk about some practical issues in our algorithm, and summarize the whole iterative framework of pronunciation selection.

4.1. A pronunciation selection criterion based on per-utterance likelihood reduction

Given a set of pronunciation candidates for a specific word w, and the conditional likelihood τ_{uwb} (acoustic evidence) for each utterance $\mathcal{O}_u$, we want to maximize the total data likelihood over the pronunciation model $\boldsymbol{\theta}_w$ [3]:

$$\mathcal{L}(\boldsymbol{\theta}_w) = \sum_u \log\left(\sum_b \tau_{uwb}\theta_{wb}\right) \tag{1}$$

where the summation $\sum_u$ is only taken over utterances where the word w actually appears. Since maximizing this objective doesn't have a closed form solution, like [8], we use EM which maximizes the following auxiliary function instead (n stands for the iteration index, $\lambda^n_{uwb} \triangleq p(w, b|\mathcal{O}_u, \boldsymbol{\theta}^n_{\boldsymbol{w}})$ is the pronunciation posterior computed at the nth iteration)

$$Q(\boldsymbol{\theta}^{n+1}_w, \boldsymbol{\theta}^n_w) = \sum_u \sum_b \lambda^n_{uwb} \log \theta^{n+1}_{wb} \tag{2}$$

Maximizing the above function with the constraint $\sum_b \theta^{n+1}_{wb} = 1$ gives the M-step:

$$\theta^{n+1}_{wb} \leftarrow \frac{\sum_u \lambda^n_{uwb}}{\sum_u \sum_b \lambda^n_{uwb}} \tag{3}$$

According to Bayes' rule, we compute the updated posteriors λ^{n+1}_{uwb} as the following:

$$\lambda^{n+1}_{uwb} \leftarrow \frac{\tau_{uwb}\theta^{n+1}_{wb}}{\sum_b \tau_{uwb}\theta^{n+1}_{wb}} \tag{4}$$

[1] we assume that each word appears in each utterance's transcript at most once. In practice, if a word appears multiple times in an utterance, we divide the utterance into sub-utterances where each one only contains one token of the word.

[2] Strictly speaking, Bayes' rule only gives us $\tau_{uwb} \propto \gamma_{uwb}$, i.e. τ_{uwb} can only be treated as γ_{uwb} up to a constant, but the constant doesn't affect the objective (1) we want to optimize.

[3] When we optimize the pronunciation probabilities for a specific word, we consider the pronunciation probabilities for other words as fixed.

which is the E-step. By running (3) and (4) iteratively until convergence, we can find an optimal pronunciation model $\boldsymbol{\theta}_w^*$, and evaluate the optimal log-likelihood (1) $\mathcal{L}(\boldsymbol{\theta}_w^*)$ (denoted as $\mathcal{L}^*$ for simplicity). In order to evaluate the importance of a specific pronunciation, say, b, we remove b from the pronunciation candidate set $\mathcal{B}$, re-initialize the pronunciation model $\boldsymbol{\theta}_w^{'}$ on top of $\mathcal{B}\backslash b$, and run EM to optimize (1) with the model $\boldsymbol{\theta}_w^{'}$. Writing the likelihood at convergence after removing b as $\mathcal{L}_b^*$, we can compute the per-utterance likelihood reduction associated with the pronunciation b as:

$$\overline{\Delta\mathcal{L}}_b \triangleq \frac{\Delta\mathcal{L}_b}{M_w} = \frac{\mathcal{L}^* - \mathcal{L}_b^*}{M_w},$$

This metric reflects the contribution of each pronunciation to the total data likelihood. With this metric, we can iteratively remove least important pronunciations in a greedy fashion, which is efficient. The complete iterative framework is given in Section 4.4.

4.2. An illustrative example

Here we show an example to illustrate the advantage of pronunciation selection based on the per-utterance log likelihood reduction $\overline{\Delta\mathcal{L}}_b$ over the learned pronunciation probabilities $\boldsymbol{\theta}_w^*$, in terms of dealing with confusability of pronunciation variants.

In Table 1, we listed the pronunciation candidates, average pronunciation posteriors, learned pronunciation probabilities, and the per-utterance log likelihood reduction of two English words 'machine' and 'us' taken from the TED-LIUM [15] training corpus. Note that the two pronunciations of 'machine' only differ in one vowel, while the two pronunciations of 'us' represent two distinct meanings.

We want a selection criterion under which it's possible to put a threshold to rule out the reduction 'M IH SH IY N' (generated from phonetic-decoding) in the 'machine' case, while keeping the acronym 'Y UW EH S' in the 'us' case. Looking at the learned pronunciation probabilities $\boldsymbol{\theta}_w^*$, it gives lower values for 'Y UW EH S' than 'M IH SH IY N', and thereby cannot serve as the criterion we need. However, the per-utterance log likelihood reduction $\overline{\Delta\mathcal{L}}_b$ of 'AH S' is much larger than 'M IH SH IY N' (0.034 v.s. 0.004). Thus it's possible to set a proper threshold on $\overline{\Delta\mathcal{L}}_b$ to keep 'AH S' and remove 'M IH SH IY N'.

The underlying reason is that the confusability between pronunciations is reflected in the sharpness of the per-utterance pronunciation posteriors γ_{uwb}. In the 'us' case, the two pronunciation variants cannot easily model each other, and therefore the posteriors are very sharp for most examples. Thereby removing the minor pronunciation 'Y UW EH S' would result in a greater reduction in the data likelihood. Thus, beyond reflecting the relative frequency, the proposed criterion $\overline{\Delta\mathcal{L}}_b$ is capable of modeling the confusability between pronunciation candidates, which is preferable from the Maximum Likelihood point of view and therefore could help us to select an informative set of pronunciations.

4.3. Refining the pronunciation selection criterion $\overline{\Delta\mathcal{L}}_b$

One difficulty of directly using $\overline{\Delta\mathcal{L}}_b$ in an iterative pronunciation selection framework is that, we need to develop an interpretable threshold T in order to decide when to stop removing pronunciations. However, we notice the upper bound of $\overline{\Delta\mathcal{L}}_b$ can be achieved in an extreme case, where we remove an absolutely dominating pronunciation p (meaning: the observed conditional likelihoods satisfy: $\tau_{uwp} = 1, \quad \tau_{uwb} = \delta, \quad \forall b \neq p$). Before removing p, it's obvious from (1) that the maximum $\mathcal{L}(\boldsymbol{\theta}_w^*) = 0$ can be reached with $\boldsymbol{\theta}_w^*$ being a one-hot vector s.t. $\theta_{wp} = 1$. After removing p, with the constraint $\sum_{b\in\mathcal{B}\backslash p}\theta_{wb}^{'} = 1$, the log-likelihood is a constant: $\mathcal{L}(\boldsymbol{\theta}_w^{'}) \equiv M_w \log\delta$. Then we have: $\overline{\Delta\mathcal{L}}_p = (0 - M_w\log\delta)/M_w = -\log\delta$. According to this, we scale this upper bound by a scalar α between $[0, 1]$ to get an interpretable threshold: $T = -\alpha\log\delta$, where $\alpha = 1$ corresponds to the above extreme case, which means, for a pronunciation to be not removed, it would have to be present with probability 1 in 100% instances of the word, and $\alpha = 0$ means we will never remove any pronunciation candidates. In practice, it's set between 0.005 and 0.2. We also make α dependent on the source $s(b)$ of the pronunciation, which enables us to use a more conservatively threshold for selecting pronunciations from a source where the candidates' quality is lower in general, like phonetic-decoding (pd), e.g. by setting $\alpha_{g2p} = 0.02, \alpha_{pd} = 0.01$. So, we define the "score" of a pronunciation candidate as "how far away" its $\overline{\Delta\mathcal{L}}_b$ is to the corresponding threshold, i.e.:

$$q_b \triangleq \overline{\Delta\mathcal{L}}_b - T_{s(b)} = \frac{\Delta\mathcal{L}_b}{M_w + \beta_{s(b)}} + \alpha_{s(b)}\log\delta$$

In our framework we iteratively prune the pronunciation with the lowest score and terminate pruning when all pronunciation have positive scores. Note that the count M_w is smoothed with a source-dependent scalar $\beta_{s(b)}$ (5-15 in practice). The purpose is to keep the score from being to high when M_w is small, so that in general we select fewer pronunciations if we only have a few acoustic examples of a word.

Table 1: *The pronunciation candidate set $\mathcal{B}$, learned pronunciation probabilities $\boldsymbol{\theta}_w^*$, and the per-utterance log likelihood reduction $\overline{\Delta\mathcal{L}}_b$ for two English words 'machine' and 'us' from TED-LIUM.*

w	'machine'	'us'
$\mathcal{B}$	['M AH SH IY N', 'M IH SH IY N']	['AH S', 'Y UW EH S']
$\boldsymbol{\theta}_w^*$	[0.987, 0.013]	[0.992, 0.008]
$\overline{\Delta\mathcal{L}}_b$	[3.575, 0.004]	[15.576, 0.034]

4.4. Summary: an iterative framework

The proposed pronunciation selection algorithm, which iteratively prunes pronunciations from the initial candidate set $\mathcal{B}$, is summarized as Algorithm 1 ($\mathcal{B}_t$ stands for the selected subset of pronunciation candidates at iteration t).

5. Experiments

In order to evaluate the performance of the proposed lexicon learning framework, a small seed lexicon is built by randomly sampling a small portion (5%) of words from the vocabulary of the expert lexicon of each task. With the seed lexicon, we train a G2P model using Sequitur [5] and apply it to all OOV (w.r.t the seed lexicon) words in the vocabulary of the expert lexicon, to get the "G2P-extended" lexicons.[4] A baseline system called G2P-ext is built using a G2P-extended lexicon with the optimal number of variants per-word tuned on dev data, and another baseline system called G2P-1best is built using a G2P-extended lexicon where we only take the top G2P pronunciation

[4] In this paper we focus on lexicon learning for alphabetic languages. Thereby a G2P model trained with a small seed lexicon is able to generate pronunciations for most words in the expert lexicon.

Algorithm 1 Greedy pronunciation selection

set $t = 0$, $\mathcal{B}_0 = \mathcal{B}$.
While true:
  Initialize $\boldsymbol{\theta}_w$ uniformly on $\mathcal{B}_t$.
  Run **EM** on $\mathcal{B}_t$ to get $\boldsymbol{\theta}_w^*$ and the optimal log-likelihood $\mathcal{L}^*$.
  For b in $\mathcal{B}_t$:
    Initialize $\boldsymbol{\theta}_w^{'}$ on $\mathcal{B}_t \backslash b$ and run **EM** to get the optimal log-likelihood $\mathcal{L}_b^*$.
    Compute $\Delta\mathcal{L}_b = \mathcal{L}^* - \mathcal{L}_b^*$
    Compute $q_b = \frac{\Delta\mathcal{L}_b}{M_w + \beta_{s(b)}} + \alpha_{s(b)} \log \delta$
  If $\min_{b \in \mathcal{B}_t} q_b \geq 0$:
    Output $\mathcal{B}_t$ as the optimal pronunciation subset.
    Break.
  Else:
    $\hat{b} = \arg\min_{b \in \mathcal{B}_t} q_b$:
  $\mathcal{B}_{t+1} = \mathcal{B}_t \backslash \hat{b}$
  $t = t + 1$

for each word. With this G2P model and acoustic training data for each task, we can build a learned lexicon using the proposed framework, and then train an ASR system called "Lex-learn". Besides, we have an ASR system trained using the full expert lexicon as the "Oracle" system. Note that the training recipes of three ASR systems (G2P-ext, G2P-1best, Oracle, and Lex-learn) for each task only differ in the lexicons (with the same vocabulary). All experiments were done with Kaldi [16].

Table 2: *ASR Performance on Librispeech (WER on the test-clean set (tuned on WER of LF-MMI systems on dev-clean, without 4-gram LM rescoring) with different lexicon conditions (the average # pronunciations per word for in-vocab words from acoustic training data, are shown in parentheses). The vocab of the full expert lexicon (a subset of CMUDict) has 200K words.*

	WER			
	Oracle (1.08)	G2P-ext (5.05)	G2P-1best (1)	Lex-learn (1.42)
SAT	11.32 %	13.11 %	14.57 %	11.53 %
LF-MMI	6.44 %	6.76 %	7.15 %	6.64 %

We conduct experiments on the Librispeech-460 task [17]. For each lexicon condition, we use the 460h training data subset to build speaker-adaptive trained GMM (SAT) models (the same AM training recipe as the "SAT 460" from [17]), on top of which we then train sub-sampled time-delay neural networks (TDNNs) [18] with the lattice-free MMI (LF-MMI) [19] criterion. The WERs are shown in Table 2. It can be seen that the learned lexicon performs better than G2P-extended lexicons, and is close to the oracle lexicon. And the LF-MMI systems are much more robust to the lexicon quality than SAT systems, i.e. the G2P-extended and learned lexicons perform closer to the expert lexicon. The learned lexicon closes 88% (SAT)/ 36%(LF-MMI) of the WER gap between the G2P-ext system and the oracle system. Also, looking at the average number of pronunciations per word, the learned lexicon (1.42) is much more compact than the G2P-extended lexicon (5.05), and is very close to the G2P-1best lexicon (1), though it performs much better than the G2P-1best lexicon by a large gap: 20.9% (SAT) / 7.1%(LF-MMI) relatively in WER.

In Table 3, we compare the proposed framework with more baseline lexicon expansion approaches, on the Librispeech-460 task (WER of SAT systems), with a smaller seed lexicon containing only $1\%(2K)$ randomly sampled words from the same expert lexicon, in order to make the performance gap between different systems more noticeable. "G2P-ext", as described before, is a baseline built with a G2P-extended lexicon (with a tuned size). "*pp*-based selection on G2P candidates" means, we first align acoustic training data with a large G2P-extended lexicon containing all G2P generated candidates (up to 10 candidates per word), and then use max-normalized pronunciation probabilities [11] to prune those candidates for each OOV word, with a tuned threshold (0.4). The pronunciation candidate pool here is the same as the G2P-ext system (i.e. G2P candidates only). "*pp*-based selection on G2P+PD candidates" uses the same lexicon expansion approach as the former one but we also add candidates from phonetic decoding (PD) before selection. Therefore this baseline has the same candidate pool as the proposed framework. The last system "likelihood-reduction-based selection on G2P+PD candidates" is the proposed framework (i.e. the "Lex-learn" systems listed before). For fair comparison, under different lexicon conditions, the acoustic models were re-trained on top of the same acoustic model (the one used in the shown G2P-ext system). It can be seen that adding PD candidates to the candidate pool is crucial to the lexicon quality (0.82% WER improvement), and the proposed pronunciation selection method solely brings 0.18% WER gain and lowers the number of pronunciations per word from 5.43 to 1.59.

Table 3: *ASR performance (WER of SAT systems on the test-clean set, without 4-gram LM rescoring) comparison on Librispeech, with different lexicon expansion approaches.*

Lexicon condition (avg. #pronunciations per word)	WER
G2P-ext (6.57)	13.72 %
pp-based selection on G2P candidates (3.77)	13.06 %
pp-based selection on G2P+PD candidates (5.43)	12.24 %
likelihood-reduction-based selection on G2P+PD candidates (1.59)	12.06 %

6. Conclusion and future work

In this paper, we propose an acoustic-data driven lexicon learning framework using a likelihood-reduction based criterion for selecting pronunciation candidates from multiple sources, i.e. G2P and phonetic decoding. With the proposed criterion, the pronunciation candidates are pruned iteratively in a greedy way, based on the acoustic data likelihood reduction caused by removing each candidate. This approach enables us to construct a compact yet informative lexicon. Experiments on different ASR tasks show that, with the proposed framework, starting with a small expert lexicon (containing $0.88K$ to$10K$ words), we are able to learn a lexicon which performs closer to a full expert lexicon in terms of WER performance on test data, than lexicons built using G2P alone or with a pruning criterion based on pronunciation probabilities. As future work, we'd like to investigate how the amount of training data affects the lexicon learning performance.

7. References

[1] M. J. Gales, K. M. Knill, and A. Ragni, "Unicode-based graphemic systems for limited resource languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5186–5190.

[2] D. F. Harwath and J. R. Glass, "Speech recognition without a lexicon-bridging the gap between graphemic and phonetic systems." in *INTERSPEECH*, 2014, pp. 2655–2659.

[3] C.-y. Lee, T. J. O'Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[4] C.-y. Lee, Y. Zhang, and J. R. Glass, "Joint learning of phonetic units and word pronunciations for asr." in *EMNLp*, 2013, pp. 182–192.

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

[6] L. Lu, A. Ghoshal, and S. Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 374–379.

[7] N. Goel, S. Thomas, M. Agarwal, P. Akyazi, L. Burget, K. Feng, A. Ghoshal, O. Glembek, M. Karafiát, D. Povey *et al.*, "Approaches to automatic lexicon learning with limited training examples," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5094–5097.

[8] I. McGraw, I. Badr, and J. 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.

[9] R. Rasipuram *et al.*, "Combining acoustic data driven g2p and letter-to-sound rules for under resource lexicon generation," in *Proceedings of INTERSPEECH*, no. EPFL-CONF-192596, 2012.

[10] A. Laurent, S. Meignier, T. Merlin, P. Deléglise, and F. Spécinov-Trélazé, "Acoustics-based phonetic transcription method for proper nouns." in *INTERSPEECH*, 2010, pp. 2286–2289.

[11] G. Chen, D. Povey, and S. Khudanpur, "Acoustic data-driven pronunciation lexicon generation for logographic languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5350–5354.

[12] S. Tsujioka, S. Sakti, K. Yoshino, G. Neubig, and S. Nakamura, "Unsupervised joint estimation of grapheme-to-phoneme conversion systems and acoustic model adaptation for non-native speech recognition," *Interspeech 2016*, pp. 3091–3095, 2016.

[13] T. Hain, "Implicit pronunciation modelling in asr," in *ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*, 2002.

[14] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian *et al.*, "Generating exact lattices in the wfst framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4213–4216.

[15] A. Rousseau, P. Deléglise, and Y. Estève, "Ted-lium: an automatic speech recognition dedicated corpus." in *LREC*, 2012, pp. 125–129.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," *Proc. ASRU*, 2011.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." in *INTERSPEECH*, 2015, pp. 3214–3218.

[19] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," *Submitted to Interspeech*, 2016.