ACOUSTIC DATA-DRIVEN PRONUNCIATION LEXICON FOR LARGE VOCABULARY SPEECH RECOGNITION

Liang Lu, Arnab Ghoshal, and Steve Renals

Centre for Speech Technology Research, University of Edinburgh, Edinburgh, EH8 9AB, UK
{liang.lu, a.ghoshal, s.renals}@ed.ac.uk

ABSTRACT

Speech recognition systems normally use handcrafted pronunciation lexicons designed by linguistic experts. Building and maintaining such a lexicon is expensive and time consuming. This paper concerns automatically learning a pronunciation lexicon for speech recognition. We assume the availability of a small seed lexicon and then learn the pronunciations of new words directly from speech that is transcribed at word-level. We present two implementations for refining the putative pronunciations of new words based on acoustic evidence. The first one is an expectation maximization (EM) algorithm based on weighted finite state transducers (WFSTs) and the other is its Viterbi approximation. We carried out experiments on the Switchboard corpus of conversational telephone speech. The expert lexicon has a size of more than 30,000 words, from which we randomly selected 5,000 words to form the seed lexicon. By using the proposed lexicon learning method, we have significantly improved the accuracy compared with a lexicon learned using a grapheme-to-phoneme transformation, and have obtained a word error rate that approaches that achieved using a fully handcrafted lexicon.

***Index Terms*—** Lexical modelling, Probabilistic pronunciation model, Automatic speech recognition.

1. INTRODUCTION

Training a speech recognition system relies on three principal resources: transcribed acoustic training data; text data for language model estimation; and a pronunciation lexicon that maps a word to one or more phonemic transcriptions. Obtaining such resources usually requires significant manual intervention, making the development of a speech recogniser for a new language, or a new domain, a costly process. The development of speech recognition systems for languages or domains with limited resources has become a major research focus in the past few years. Encouraging results have been reported for acoustic modelling using very limited amounts of transcribed audio for a new target language, e.g. by leveraging the acoustic data of other languages using multi-layer perceptrons [1, 2] or subspace Gaussian mixture models [3, 4].

The pronunciation lexicon usually has a large, finite vocabulary and is handcrafted by linguistic expert. Building a lexicon is normally time consuming and expensive; moreover, the expert pronunciation lexicon is usually fixed during the training and application of an ASR system. Updating the lexicon to cover additional words is not a trivial task. Automatically learning the pronunciation lexicon has been pursued for more than a decade, with the original focus being on the learning of pronunciation variations or alternative pronunciations for some words [5, 6, 7, 8]. More recently, McGraw, et al. [9] proposed a stochastic pronunciation mixture model framework to automatically update the pronunciation weights of the words in the lexicon, Such approaches assume the availability of a high-quality initial pronunciation dictionary and aim to add suitable pronunciation variations of words beyond the canonical pronunciations present in the dictionary.

There have been previous attempts to move beyond phonemic baseforms and jointly learn the inventory of subword units and the pronunciation lexicon. For instance, Bacchiani and Ostendorf [10] proposed an iterative acoustic segmentation and clustering approach to build a sub-word inventory from the acoustics and then automatically construct a dictionary based on those sub-word units. Later on, Singh, et al. [11] presented an expectation-maximisation (EM) algorithm for this purpose and demonstrated some promising results on the relatively small resource management (RM) corpus.

While learning the entire lexicon from scratch is challenging for large vocabulary speech recognition task, a more practical technique is to enlarge an expert phonemic lexicon by learning the pronunciations of additional words and update the acoustic model based on this updated lexicon. This approach was used in learning pronunciations of names [12, 13], or learning pronunciations of all types of words starting from a small seed lexicon [14, 15]. The work reported in this paper follows this general approach. We start with a seed lexicon and use a grapheme-to-phoneme (G2P) converter [16] to generate multiple pronunciations for new words. We present a WFST-based EM algorithm to estimate the weights of these alternative pronunciations based on acoustic evidence, and compare with its Viterbi approximation. On the 300-hour Switchboard task we start with a random 5000-word subset of the 30,000-word expert lexicon and show that the proposed method is able to learn the pronunciations of the remaining words with only a small reduction in accuracy.

2. PROBABILISTIC PRONUNCIATION MODEL

Given the acoustic observations $\mathbf{O}$, the optimal word sequence $\hat{\mathbf{W}}$ is obtained from a conventional ASR engine as

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \; p(\mathbf{O}|\mathcal{M}, \mathbf{W})P(\mathbf{W}), \tag{1}$$

where $\mathcal{M}$ denotes the acoustic model parameters, and $P(\mathbf{W})$ is the prior probability for word sequence $\mathbf{W}$, normally obtained from the language model. In order to improve the modelling accuracy as well as to generalise to unseen words in the training dataset, the acoustic model $\mathcal{M}$ usually operates at the level of sub-word units (typically context-dependent phonemes) rather than words. The mapping from each word to its corresponding phonemic transcription is usually defined by a handcrafted pronunciation lexicon in which most of the words have a single canonical pronunciation.

This work was funded by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

When the dictionary contains multiple pronunciations per word with corresponding pronunciation weights that form a valid probability distribution, the decision rule (1) may be written as:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \ P(\mathbf{W}) \sum_{\mathbf{B} \in \Psi_{\mathbf{W}}} p(\mathbf{O}|\mathcal{M}, \mathbf{B}) P(\mathbf{B}|\mathbf{W}), \qquad (2)$$

where $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ denotes a valid pronunciation sequence for the word transcription $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$, and $P(\mathbf{B}|\mathbf{W})$ denotes its probability. $\mathbf{b}_i$ is the pronunciation of word $\mathbf{w}_i$. $\Psi_{\mathbf{W}}$ denotes the set of all the possible pronunciation sequences of $\mathbf{W}$. This is same as the formulation used in [12, 13], and in [9] where the dictionary is viewed as containing all possible *phonetic* realisations of a word and the model is referred to as a pronunciation mixture model.

The pronunciation of words may depend on the surrounding words. However a reasonable simplification is to ignore this effect and express $P(\mathbf{B}|\mathbf{W})$ as the product of the pronunciation probabilities of each word:

$$P(\mathbf{B}|\mathbf{W}) = P(\mathbf{b}_1|\mathbf{w}_1) \cdots P(\mathbf{b}_n|\mathbf{w}_n) \ . \qquad (3)$$

Like in [9], we assume that each word may have multiple surface pronunciations with a corresponding probability weight. More formally, it may be expressed as

$$P(\mathbf{b}_i = \mathbf{p}_j|\mathbf{w}_i) = \theta_{ij}, \quad j = 1, \ldots, J_i \qquad (4)$$

$$\text{subject to:} \quad \sum_j \theta_{ij} = 1 \ . \qquad (5)$$

where J_i is the number of alternate pronunciations of $\mathbf{w}_i$, and $\mathbf{p}_j$ denotes one of those surface pronunciations with a weight θ_{ij}.

2.1. Pronunciation weight estimation using EM

Given a pronunciation dictionary, an acoustic model $\mathcal{M}$, and some transcribed acoustic data, the pronunciation weights θ_{ij} can updated using the EM algorithm [11, 12, 13, 9]. If $\mathbf{O}_r$ represents the acoustic observations and $\mathbf{W}_r$ the corresponding word-level transcription for $r = 1, \ldots, R$ training utterances, then the pronunciation weight θ_{ij} for a particular word and pronunciation pair $(\mathbf{w}_i, \mathbf{p}_j)$ can be optimized using the following EM auxiliary function:

$$\begin{aligned} \mathcal{Q}(\theta_{ij}) &= \sum_{r=1}^{R} \sum_{\mathbf{B}_r \in \Psi^*_{\mathbf{W}_r}} P(\mathbf{B}_r|\mathbf{O}_r, \mathcal{M}, \mathbf{W}_r) \log P(\mathbf{B}_r|\mathbf{W}_r) + k \\ &= \sum_{r=1}^{R} \sum_{\mathbf{B}_r \in \Psi^*_{\mathbf{W}_r}} P(\mathbf{B}_r|\mathbf{O}_r, \mathcal{M}, \mathbf{W}_r) C_{ij|\mathbf{B}_r} \log \theta_{ij} + k \end{aligned} \qquad (6)$$

where $C_{ij|\mathbf{B}_r}$ denotes the number of times that $(\mathbf{w}_i, \mathbf{p}_j)$ appears in the pronunciation sequence $\mathbf{B}_r$. $\Psi^*_{\mathbf{W}_r}$ represents the subset of pronunciation sequences that contain $(\mathbf{w}_i, \mathbf{p}_j)$, and k is a constant which does not depend on θ_{ij}. The posterior probability of the pronunciation sequence $\mathbf{B}_r$ can be computed according to Bayes' rule:

$$P(\mathbf{B}_r|\mathbf{O}_r, \mathcal{M}, \mathbf{W}_r) = \frac{p(\mathbf{O}_r|\mathbf{B}_r, \mathcal{M}) P(\mathbf{B}_r|\mathbf{W}_r)}{\sum_{\mathbf{B}_r \in \Psi_{\mathbf{W}_r}} p(\mathbf{O}_r|\mathbf{B}_r, \mathcal{M}) P(\mathbf{B}_r|\mathbf{W}_r)},$$

where $P(\mathbf{B}_r|\mathbf{W}_r)$ is calculated using the old estimate of the pronunciation weights. Using this, the auxiliary function $\mathcal{Q}(\theta_{ij})$ may be rewritten as

$$\mathcal{Q}(\theta_{ij}) = \sum_{r=1}^{R} \lambda_{ijr} \log \theta_{ij} + k \qquad (7)$$

where λ_{ijr} represents the following term:

$$\lambda_{ijr} = \frac{\sum_{\mathbf{B}_r \in \Psi^*_{\mathbf{W}_r}} p(\mathbf{O}_r|\mathbf{B}_r, \mathcal{M}) P(\mathbf{B}_r|\mathbf{W}_r) C_{ij|\mathbf{B}_r}}{\sum_{\mathbf{B}_r \in \Psi_{\mathbf{W}_r}} p(\mathbf{O}_r|\mathbf{B}_r, \mathcal{M}) P(\mathbf{B}_r|\mathbf{W}_r)} \ . \qquad (8)$$

Maximising this under the constraint that $\sum_j \theta_{ij} = 1$, the new value of θ_{ij} is obtained as:

$$\hat{\theta}_{ij} = \frac{\sum_r^R \lambda_{ijr}}{\sum_{r=1}^{R} \sum_j \lambda_{ijr}} \qquad (9)$$

However, the computation of λ_{ijr} may be expensive, since the size of the pronunciation sequence space $\Psi_{\mathbf{W}_r}$ is exponential in the length of $\mathbf{W}_r$, which prohibits an exhaustive enumeration of all possible sequences. To reduce the computational requirement, [13] and [9] propose approximating $\Psi_{\mathbf{W}_r}$ with an N-best list of alternate pronunciation sequences generated by a G2P model, and rescoring this N-best list with the acoustic model. However, this is still costly due to the repeated evaluation of $p(\mathbf{O}_r|\mathbf{B}_r, \mathcal{M})$ for each pronunciation sequence $\mathbf{B}_r$ on the N-best list. Moreover, it may introduce a loss in accuracy since the search space is reduced, especially for shallower N-best lists.

Instead of N-best pronunciation sequences, we propose using speech recognition lattices. Lattices are an efficient representation of an exponential number of alternatives using linear space. Moreover, lattices only contain pronunciation variants that have sufficiently high likelihood given the acoustics and hence we do not need to rescore very unlikely pronunciation variants as in the N-best list approach. It is important to note that since we do not do an unconstrained phonetic decoding (see section 2.2), the concern about learning "linguistically incorrect pronunciations" expressed in [13] is not applicable here[1]. Another motivation for using lattices is that efficient polynomial time algorithms exist for computing posterior probabilities and the value of $C_{ij|\mathbf{B}_r}$ over lattices. In the following subsection we present a WFST-based formulation to do so.

2.2. WFST-based pronunciation weight estimation

The space of all possible pronunciation sequences for a word sequence $\mathbf{W}_r$ may be represented as a weighted finite state transducer:

$$\mathcal{P}_r = \min(\det(\mathcal{L} \circ \mathcal{G}_r)), \qquad (10)$$

where $\mathcal{L}$ is the lexicon transducer that maps the words to their corresponding pronunciations; $\mathcal{G}_r$ is a linear acceptor representing $\mathbf{W}_r$; det and min denote the determinisation and minimisation operations respectively; and $\circ$ is the FST composition operation [18]. The pronunciation variants in $\mathcal{P}_r$ are scored by running a recogniser over the decoding graph $\mathcal{D}_r = \min(\det(\mathcal{H} \circ \mathcal{C} \circ \mathcal{P}_r))$, where $\mathcal{C}$ is the context-dependency transducer and $\mathcal{H}$ represents the HMM set. The hypothesis space of the recogniser containing the likelihoods of the pronunciation variants is obtained as a phone lattice [19]. The arcs of this lattice contain both acoustic likelihoods and pronunciation weights, and the lattice is further converted such that the arc costs form a log semiring [18]. This lattice is, in practise, the $\Psi_{\mathbf{W}_r}$ in equation (8).

The denominator of (8) can be computed by using the OpenFST [20] tool `fstshortestdistance` on $\Psi_{\mathbf{W}_r}$, which computes the summation of the weights of all the paths since $\Psi_{\mathbf{W}_r}$ is in log semiring. Its computational complexity is linear in the number of states

[1] Although not relevant to the current work, such *non-canonical* pronunciations may not be a limitation as long as they are consistent [17].

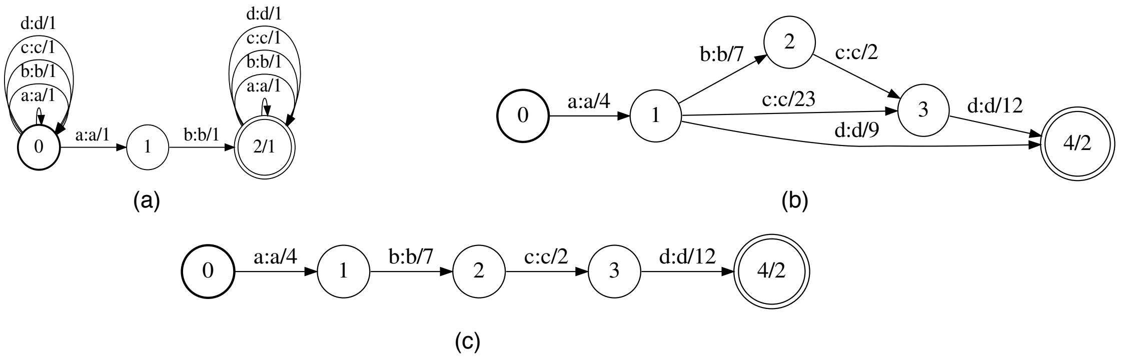

Fig. 1. A toy example for WFST-based pronunciation weight estimation, where the phone unit inventory is $\Sigma = \{a, b, c, d\}$. The transducers are shown in the form of real semiring for clarity. (a): The path counting transducer for pronunciation $\mathbf{p}_1$ = "a b" . (b): A toy example of Ψ that contains $\mathbf{p}_1$. (c): The path with its corresponding weights obtained by the composition of (a) and (b).

Table 1. Pronunciation labels using position-dependent and position-independent phones, where "_S" indicates a singleton phone, and "_B", "_I" and "_E" represent word beginning, internal and end phones, respectively.

words	pronunciation	type
a	ay ay_S	position-independent position-dependent
able	ey b ax l ey_B b_I ax_I l_E	position-independent position-dependent

and arcs for the class of acyclic transducers [18] that $\Psi_{\mathbf{W}_r}$ belongs to. To compute the numerator of equation (8), we borrow the idea of path counting transducer used in language modelling [21]. We construct a path counting transducer for each pronunciation to select the paths in $\Psi_{\mathbf{W}_r}$ containing the pronunciation and to accumulate their weights using WFST composition. This is illustrated in Figure 1.

However, this approach fails for pronunciations that are subsequences of other pronunciations. For instance, in Figure 1 if both $\mathbf{p}_1$ = "a b" and $\mathbf{p}_2$ = "a b c" are valid pronunciations for the word we are interested in, then the counts for $\mathbf{p}_1$ will also include the paths that actually contain $\mathbf{p}_2$. This is mostly remedied by simply having word-position markers on the phonemes. Examples are given in Table 1, where the suffix labels "_S", "_B", "_I" and "_E" are used to label the singleton, word beginning, word internal, and word end phones receptively. But the word position markers cannot account for the special case of *homophones*, i.e. when the subsequences are of the same length and hence identical even with position markers. It is possible to create a time-sensitive counting transducer[2], but in this work we removed the utterances containing homophones from the training data. This removed around 15–30% of training data depending on the number of alternative pronunciations for each word.

2.3. Viterbi-based pronunciation weight estimation

Another approach is to use the Viterbi approximation to update the pronunciation weights. This uses the implicit assumption that the likelihood of the most likely path is much larger than the others. The Viterbi-based estimation is similar to the method used in [14] except that there the authors retained only the most likely pronunciation for any new word whereas here we retain all pronunciations that have sufficiently high weight. The auxiliary function for the weights θ_{ij} when using the Viterbi approximation may be written as

$$\mathcal{Q}(\theta_{ij}) = \sum_{r=1}^{R} C_{ij|\hat{\mathbf{B}}_r} \log \theta_{ij} + k \tag{11}$$

$$\hat{\mathbf{B}}_r = \arg\max_{\mathbf{B}_r} P(\mathbf{B}_r|\mathbf{O}, \mathcal{M}, \mathbf{W}_r). \tag{12}$$

Then the updated value of θ_{ij} is

$$\hat{\theta}_{ij} = \frac{\sum_{r=1}^{R} C_{ij|\hat{\mathbf{B}}_r}}{\sum_{r=1}^{R} \sum_j C_{ij|\hat{\mathbf{B}}_r}}. \tag{13}$$

In this case only the most likely path needs to be obtained from the decoder instead of the lattice. The value of $C_{ij|\hat{\mathbf{B}}_r}$ can be easily obtained by aligning the word with its corresponding pronunciation according to the lexicon. Again, we use position-dependent phone labels so that we can obtain the word boundary for the alignment. Compared to the WFST-based implementation, this method has lower computation and memory usage since the WFST composition and `fstshortestdistance` operations are not required. Furthermore, it is applicable to utterances with homophones which means the entire training data can be used.

3. ITERATIVE LEARNING OF LEXICON AND ACOUSTIC MODELS

We follow an iterative training schedule, where the pronunciation weights are estimated given the acoustic model and the acoustic model is re-estimated given the newly selected set of pronunciations. This alternating training schedule is commonly used in literature [11, 10, 14] instead of direct joint estimation, which is computationally cumbersome besides being unlikely to provide any accuracy gains. Starting from a seed lexicon, we train a grapheme-to-phoneme (G2P) model [16] to generate multiple alternative pronunciations of the words in the training set. The pronunciation weights are initialised to be uniform and are then updated using the WFST- or Viterbi-based estimation described in the preceding subsections. A few examples of words with their learned pronunciations and weights can be seen in Table 2.

[2] This can be implemented using a specialized `Matcher` class in OpenFST that uses word-timing information.

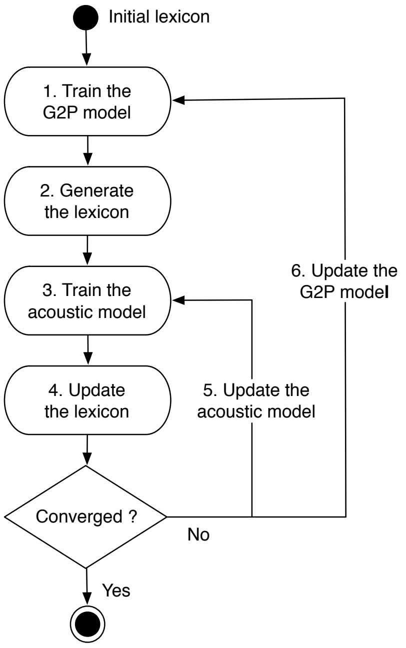

Fig. 2. An iterative training scheme to learn the lexicon and acoustic models.

With multiple pronunciations per word, the EM estimation of the acoustic model $\mathcal{M}$ maximises the following objective:

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \sum_{r=1}^{R} \log \sum_{\mathbf{B}_r \in \Psi_{\mathbf{W}_r}} p(\mathbf{O}_r | \mathcal{M}, \mathbf{B}_r) P(\mathbf{B}_r | \mathbf{W}_r), \quad (14)$$

where the sufficient statistics are accumulated from all the possible pronunciation sequences $\mathbf{B}_r \in \Psi_{\mathbf{W}_r}$ for each utterance $\mathbf{W}_r$. However, to simplify the implementation and to reduce the computational cost, the Viterbi approximation is used in practise where the sufficient statistics are accumulated only from the most likely pronunciation sequence:

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \max_{\mathbf{B}_r} \sum_{r=1}^{R} p(\mathbf{O}_r | \mathcal{M}, \mathbf{B}_r) P(\mathbf{B}_r | \mathbf{W}_r) \ . \quad (15)$$

To sum up, the recipe used in this paper is shown in Figure 2 and is detailed as follows:

1. Train the G2P model using the available initial seed lexicon.
2. For each word in the training transcriptions, generate up to N pronunciations using the G2P model.
3. Train the acoustic model using the current lexicon.
4. Update the pronunciation weights using the algorithm described in Section 2 and prune those pronunciations whose weights are below a threshold $\theta_{ij} \leq T_1$.
5. Go to step 3 to re-train the acoustic model with the new lexicon until the maximum number of iterations is reached.
6. Go to step 1 to re-train the G2P model with those pronunciations whose weight is above the threshold $\theta_{ij} \geq T_2$ until the maximum number of iterations is reached.

In the experiments described below, we will show that only a few iterations of joint acoustic and lexicon model training is sufficient for the system to converge according to this recipe.

Table 2. Examples of pronunciation learning using the probabilistic pronunciation model, which are shown using position independent phone unit for clarity.

Word	Initial		Updated	
	Pronunciations	θ	Pronunciations	θ
abilities	ae b ih l ih t iy z	0.2	ax b ih l ih t iy z	1.0
	ae ey b ih l ih t iy z	0.2		
	ax b ih l ih t iy z	0.2		
	ay ax b ih l ih t iy z	0.2		
	s ax b ih l ih t iy z	0.2		
aboveboard	ax b ah ax v b ow r d	0.2	ax b ah ax v b ow r d	0.62
	ax b ah v b ow r d	0.2	ax b ah v b ow r d	0.38
	ax b ah v ey b ow r d	0.2		
	ax b ah v l b ow r d	0.2		
	ax b ah v r b ow r d	0.2		
instance	ih n s t ae n s	0.2	ih n s t ax n s	0.72
	ih n s t ae n s ih	0.2	ih n s t ih n s	0.28
	ih n s t ax n s	0.2		
	ih n s t en s	0.2		
	ih n s t ih n s	0.2		

4. EXPERIMENTS AND RESULTS

We performed experiments on the Switchboard corpus, where the training set contains about 300 hours of conversational telephone speech. The Hub-5 Eval 2000 data is used as the test set. We used the Kaldi speech recognition toolkit [22] to train conventional GMM-based acoustic models following the recipe described in [23]. 39-dimensional MFCC+Δ + $\Delta\Delta$ features were used, using a context window of 7 frames to which linear discriminant analysis transformations were then applied to reduce the feature dimensionality to be 40, followed by a global semi-tied covariance matrix transform [24] to decorrelate the features. The expert lexicon was supplied by the Mississippi State transcriptions and it has more than 30,000 words, of which a random subset of 5,000 words is used as the seed lexicon.

4.1. Results: WFST-based training

We first evaluate WFST-based training for lexicon learning. The initial benchmarking and tuning experiments were performed using a 110 hour subset of the Switchboard corpus, which has a vocabulary size of about 20,000 words. The acoustic model has around 3,900 tied triphone states and 90,000 Gaussian components overall. The systems were trained using the maximum likelihood (ML) criterion without speaker adaptation. The baseline system using the expert lexicon achieves a recognition word error rate (WER) of 37.2%. It is not feasible to build a system using the seed lexicon solely, since most of the words in the training data are missing from the vocabulary. To tackle this problem, we used the toolkit described in [16] and trained a G2P model using the 5,000 entries in the seed lexicon. We then generated the 1-best pronunciations for the missing words, and built a baseline system which has a WER of 42.1%. This system provides a reasonable baseline for the lexicon learning experiments.

Following the recipe in Section 3, we learned the pronunciations from the training data using the WFST-based training approach. We first used the G2P model described above to generate $N = 5$ pronunciations for each word, with all pronunciations for a word initialised to have equal weights. With the pronunciation dictionary initialised in this way, we obtained 47.0% WER, which is much higher than the G2P baseline due to the large number of alternative pronunciations which cause confusions. We then performed the iterative lexicon and acoustic model learning algorithm, and pruned those pronunciations

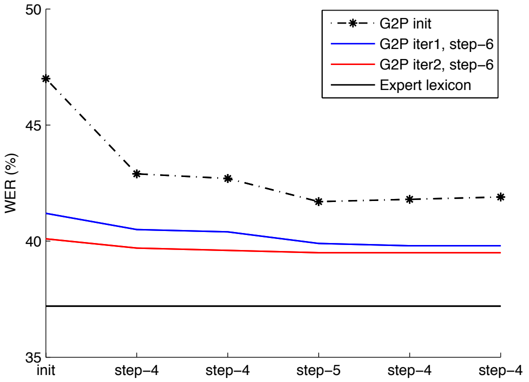

Fig. 3. Results of WFST-based system using the 110 hours training data.

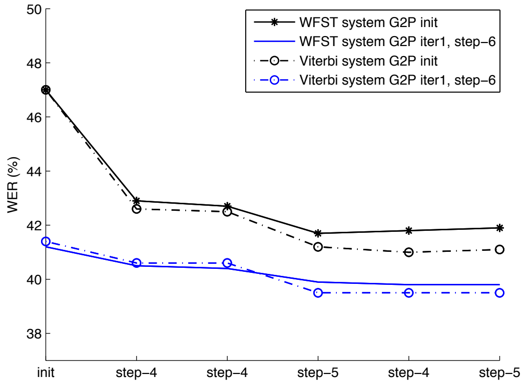

Fig. 4. Comparison of WFST- and Viterbi-based systems using the 110 hours of training data.

whose weights were below the threshold $T_1 = 0.1$. After the model converged, the system achieved 41.7% WER which is slightly better than the G2P baseline.

Given the output of the new lexicon, we then selected those pronunciations with weights above the threshold $T_2 = 0.4$, and used them together with the seed lexicon to form a new training set to update the G2P model. This improved the accuracy of the G2P model. Following another two iterations of lexicon and acoustic model learning, we obtained a significantly reduced WER of 39.5%. Detailed results are shown in Figure 3. We observed that the training converges after 2 iterations. This may be due to the fact that we set an aggressive pruning threshold, $T_1 = 0.1$. However, using a lower threshold did not improve the accuracy but did make training a bit slower. Table 2 shows a few examples of the pronunciations that learned by the WFST-based system. Words seen in the training data have around 1.2 pronunciations on average after the training converges. With a lower threshold value, the average number of pronunciations is larger, however, it does not reduce the word error rate since it causes more pronunciation confusions. Using a discriminative learning criterion to learn pronunciations may prove helpful in this case [25, 13].

Table 3. Comparison of WFST- and Viterbi-based systems with and without filtering the utterances with homophones.

System	WER (%)
WFST system + filtered training set	42.9
Viterbi system + filtered training set	42.8
Viterbi system + whole training set	42.6

Table 4. WERs (%) of the systems trained on 300 hours of training data and tested on the Hub-5 2000 eval set.

System	Callhome	Swb	Avg
G2P baseline (ML)	46.3	29.0	37.7
110-hr lexicon baseline (ML)	44.2	27.2	35.9
G2P iter1 (ML)	43.6	26.1	35.0
+ ML-SAT	38.2	23.2	30.8
+ bMMIE-SAT	35.1	20.5	27.8
Expert lexicon baseline (ML)	42.3	25.3	34.0
+ ML-SAT	36.8	22.0	29.4
+ bMMIE-SAT	33.5	19.3	26.4

4.2. Results: Viterbi-based training

We compared the performance of the WFST-based system with a Viterbi-based system. As discussed above, the WFST-based implementation cannot distinguish homophones; in our previous experiments we filtered out those training utterances that contained homophones. However, the Viterbi-based training method does not have this problem. Table 3 shows the results of updating the initial lexicon using 1 iteration of either the WFST- or Viterbi-based algorithm. We found that when using the same amount of training data, the WFST-based system did not outperform the Viterbi-based counterpart, and when using the whole training set the Viterbi-based system achieved a slightly lower WER. Figure 4 shows more results following iterative lexicon and acoustic model learning, and we observed a similar trend. Here the Viterbi-based systems used the whole training set of the 110 hours of training data.

Next, we performed the experiments using the complete 300 hours of Switchboard acoustic training data, the results of which are presented in Table 4. We started with the lexicon learned from 110 hours of training data and retrained the acoustic model on the 300-hour training set. This is the "110-hr lexicon baseline" in the table. The refining of pronunciations based on acoustic evidence from even the 110-hour training set significantly improves on the "G2P baseline" where a G2P model trained on the 5000-word seed lexicon is used to generate a single pronunciation for all the remaining 25,000 words in the lexicon and then the acoustic model is trained on the 300-hour training set. Next, we performed another round of iterative lexicon and acoustic model update, using Viterbi training throughout ("+G2P iter1"). This lowers the WER by another 0.9% since the pronunciations of an additional 10,000 words could now be refined based on acoustic evidence. The WER of this system is only 1% higher than that of a comparable ML trained system that uses the entire 30,000-word expert lexicon.

Finally, speaker adaptive training (SAT) [26] is done using a single feature-space maximum likelihood linear regression (FMLLR) transform [27] estimated per speaker, and discriminative training using boosted MMIE (bMMIE) [28]. We find the WER gap between the systems using the learned and expert lexicon widening when using SAT and bMMIE, which could be due to the fact that the lexicon is optimised for the ML-trained models.

5. CONCLUSION

This paper is about building pronunciation lexicon for speech recognition using transcribed acoustic data. This paper presents a WFST-based EM algorithm and its Viterbi approximation for estimating pronunciation weights using acoustic evidence. To demonstrate the effectiveness of the lexicon learning method, experiments were performed on the Switchboard corpus which contains around 300 hours of conversational telephone speech. The expert lexicon has about a 30,000 word vocabulary, from which randomly selected 5,000 words as our seed lexicon. By expanding the seed lexicon, we obtained a WER that approaches that obtained using the expert lexicon. A constraint of this work is the requirement of a seed lexicon; to learn a lexicon from scratch, an unsupervised initialisation method is needed which will be one of our future works.

6. REFERENCES

[1] S Thomas, S Ganapathy, and H Hermansky, "Cross-lingual and multi-stream posterior features for low resource LVCSR systems," in *Proc. INTERSPEECH*, 2010, pp. 877–880.

[2] P Swietojanski, A Ghoshal, and S Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc IEEE SLT*, 2012, p. 246251.

[3] Y Qian, D Povey, and J Liu, "State-level data borrowing for low-resource speech recognition based on subspace GMMs," in *Proc. INTERSPEECH*, 2011.

[4] L Lu, A Ghoshal, and S Renals, "Regularized subspace Gaussian mixture models for cross-lingual speech recognition," in *Proc. IEEE ASRU*, 2011, pp. 365–370.

[5] T Slobada and A Waibel, "Dictionary learning for spontaneous speech recognition," in *Proc. ICSLP*, 1996, pp. 2328–2331.

[6] M Saraclar, H Nock, and S Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech & Language*, vol. 14, no. 2, pp. 137–160, 2000.

[7] M Wester, "Pronunciation modeling for ASR — knowledge-based and data-derived methods," *Computer Speech & Language*, vol. 17, no. 1, pp. 69–85, 2003.

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

[9] I McGraw, I Badr, and J Glass, "Learning lexicons form speech using a pronunciation mixture model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 357–366, 2013.

[10] M Bacchiani and M Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, no. 2, pp. 99–114, 1999.

[11] R Singh, B Raj, and RM Stern, "Automatic generation of subword units for speech recognition systems," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 2, pp. 89–99, 2002.

[12] F Beaufays, A Sankar, S Williams, and M Weintraub, "Learning name pronunciations in automatic speech recognition systems," in *Proc. ICTAI*. IEEE, 2003, pp. 233–240.

[13] X Li, A Gunawardana, and A Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proc. ASRU*. IEEE, 2007, pp. 130–135.

[14] N Goel, S Thomas, M Agarwal, P Akyazi, L Burget, K Feng, A Ghoshal, O Glembek, M Karafiát, D Povey, A Rastrow, RC Rose, and P Schwarz, "Approaches to automatic lexicon learning with limited training examples," in *Proc. ICASSP*. IEEE, 2010, pp. 5094–5097.

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

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

[17] B Hutchinson and J Droppo, "Learning non-parametric models of pronunciation," in *Proc. IEEE ICASSP*, May 2011, pp. 4904–4907.

[18] M Mohri, F Pereira, and M Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, January 2002.

[19] D Povey, M Hannemann, G Boulianne, L Burget, A Ghoshal, M Janda, Ma Karafiat, S Kombrink, P Motlicek, Y Qian, K Riedhammer, K Vesely, and NT Vu, "Generating exact lattices in the WFST framework," in *Proc. IEEE ICASSP*, March 2012, pp. 4213–4216.

[20] C Allauzen, M Riley, J Schalkwyk, W Skut, and M Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Implementation and Application of Automata*, pp. 11–23. Springer, 2007.

[21] C Allauzen, M Mohri, and B Roark, "Generalized algorithms for constructing statistical language models," in *Proc. ACL*. Association for Computational Linguistics, 2003, pp. 40–47.

[22] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlıcek, Y Qian, P Schwarz, J Silovský, G Semmer, and K Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[23] K Veselý, A Ghoshal, L Burget, and D Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013.

[24] MJF Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[25] O Vinyals, L Deng, D Yu, and A Acero, "Discriminative pronounciation learning using phonetic decoder and minimum-classification-error criterion," in *Proc. ICASSP*. IEEE, 2009, pp. 4445–4448.

[26] Y Anastasakos, J McDonough, R Schwartz, and J Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996.

[27] MJF Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[28] D Povey, D Kanevsky, B Kingsbury, B Ramabhadran, G Saon, and K Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. ICASSP*. IEEE, 2008, pp. 4057–4060.