On Mispronunciation Lexicon Generation using Joint-sequence Multigrams in Computer-Aided Pronunciation Training (CAPT)

Xiaojun Qian[1], *Helen Meng*[1] *and Frank Soong*[1,2]

[1]Human-Computer Communications Laboratory,
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong SAR of China
[2]Speech Group, Microsoft Research Asia, Beijing, China

`xjqian@se.cuhk.edu.hk, hmmeng@se.cuhk.edu.hk, frankkps@microsoft.com`

Abstract

We investigate the use of joint-sequence multigrams to generate L2 mispronunciation lexicons for mispronunciation detection and diagnosis. In the joint-sequence framework, a pair of parallel strings (namely, the input string of either graphemes or phonemes of the canonical pronunciation and the phonetic string of the mispronunciation) are aligned to form joint units for probabilistic estimation. We compare results on lexicons produced by phoneme-to-mispronunciation conversion and those by grapheme-to-mispronunciation conversion. Results reflect the hypothesized advantage (1.1% reduction in expected miss rate) in unifying phonetic confusion due to L1 negative transfer with those due to grapheme-to-phoneme errors. The impact of mispronunciation by mis-use of analogy is also studied. Recognition results show the benefit of a lexicon with proper priors.

Index Terms: mispronunciation detection and diagnosis, lexicon extension, joint-sequence multigrams

1. Introduction

Mispronunciation detection is an important problem in CAPT. Previous approaches include: (1) ASR-based pronunciation scoring, e.g. Goodness of Pronunciation [1]; (2) the use of acoustic-phonetic classifiers based on specific features for targeted mispronunciations [2]; (3) explicit modeling of mispronunciations for pronunciation lexicon (or network topology) extension [3], etc. The first two approaches try to make a binary decision - typically "correct" or "mispronounced", for each input L2 phone segment. Phone segments labeled as "mispronounced" may correspond to a nonnative version of the same phone, a deletion, an insertion, a substitution or even a noncategorical change. However, these approaches usually suffer from error propagation by performing a segmentation based on canonical transcriptions in the first phase. The third strand of approaches for mispronunciation detection and diagnosis are often adopted to alleviate the segmentation bias and provide additional local feedback. The idea is to model the common mispronunciation pattern by heuristics or by learning from data. Thus one can either align the L2 speech in the extended network by state-of-the-art acoustic models in a one-shot manner, or use the ASR confidence scores for further manipulation.

2. Previous Work

Our previous efforts on contrastive analysis of L2 speech are strongly inspired by concepts from generative phonology. It aims to model the *L1 negative transfer* effect by context-dependent phonological rules (**PR**): $\lambda[\phi]\rho \rightarrow \psi$, which states that the canonical phone string ϕ with left context λ and right context ρ receives the corresponding phone string ψ. These rules can either be specified by knowledge or automatically derived from data [4] based on a phonetic alignment [3]. Yet we found that the knowledge-based contrastive analysis paradigm cannot properly handle many of the mispronunciations in practical situations, e.g. those caused by *grapheme-to-phoneme errors*. In this case, even the rules extracted from data are sometimes hard to justify, bringing about a huge number of false alarms in the mispronunciation lexicon.

Our recent work proposes to apply the grapheme-to-mispronunciation (**G2M**) conversion as an alternative to PR since it better copes with the L2 mispronunciation generation process [5], instead of the phoneme-to-mispronunciation (**P2M**) correspondence assumed by PR. The generative joint-sequence multigram model (**JSM**) is trained on pairs of grapheme and phoneme strings to construct mispronunciation lexicons. Results show that the lexicon derived by the G2M JSM covers more mispronunciation variants than the PR under the same decoding network complexity [5]. Some of the aspects left unexplored for this new paradigm includes whether the improved performance is due to the joint sequence modeling of the orthographic form and their respective mispronunciations, or merely better formulations and modeling techniques of the JSM. We wish to explore the relative contributions of these two factors (P2M and G2M) causing mispronunciations in terms of improved coverage (if any) of possible learners' errors in the mispronunciation lexicon generated. Studies are needed to examine the extent to which the canonical pronunciations can lead to mispronunciations.

The remainder of this paper is organized as follows: In the next section, we review the JSM in the context of mispronunciation modeling. In Section 4, we will take a detailed exposition of the JSM for L2 segmental mispronunciations generation in various settings. Finally, conclusions are drawn and directions for future studies are proposed.

3. Joint-sequence Multi-gram Model

A joint-multigram (see Eq. (1))

$$q = (g, \varphi) \in Q \subseteq \{\{G^{\{0,1,\ldots,L_G\}} \times \Phi^{\{0,1,\ldots,L_\Phi\}}\} \backslash \{G^0 \times \Phi^0\}\} \tag{1}$$

is the pair of a discrete input string g and a discrete output string φ of possibly different lengths. L_G and L_Φ are upper limits of the lengths of input and output strings and they control the size of the joint-multigram inventory[1]. The joint-sequence multigram model assumes that each pair of input and output strings $\mathcal{O} = (\boldsymbol{g}, \boldsymbol{\varphi})$ is generated by a common sequence of joint multigrams $\boldsymbol{q}$ which uniquely defines a co-segmentation (i.e. alignment), thus enabling the application of traditional n-multigram language modeling techniques [7]. Due to the various possibilities of co-segmentations between $\boldsymbol{g}$ and $\boldsymbol{\varphi}$, the JSM determines the joint probability by summing over all matching joint multigrams:

$$p(\mathcal{O}) = p(\boldsymbol{g}, \boldsymbol{\varphi}) = \sum_{\sigma \in \{\sigma\}} p(\sigma, \mathcal{O}) = \sum_{\boldsymbol{q} \in S(\mathcal{O})} p(\boldsymbol{q}), \tag{2}$$

where σ is a co-segmentation of $\boldsymbol{g}$ and $\boldsymbol{\varphi}$, and $S(\mathcal{O})$ is the set of all co-segmentations in terms of $\boldsymbol{q}$:

$$S(\mathcal{O}) = S(\boldsymbol{g}, \boldsymbol{\varphi}) = \{\boldsymbol{q} \in Q^* | \sqcup \boldsymbol{q}_g = \boldsymbol{g}, \sqcup \boldsymbol{q}_\varphi = \boldsymbol{\varphi}\}. \tag{3}$$

Here, we use $\boldsymbol{q}_g$ and $\boldsymbol{q}_\varphi$ to denote the sequences of $\boldsymbol{q}$'s first and second component, respectively, and "$\sqcup$" denotes string concatenation. For each possible $\boldsymbol{q}$, the probability $p(\boldsymbol{q})$ can be approximated by a standard $\mathcal{M}$-gram approximation:

$$p(\boldsymbol{q}) = \prod_{j=1}^{|\boldsymbol{q}|} p(q_j | q_{j-1}, \ldots, q_{j-\mathcal{M}+1}), \tag{4}$$

where $|\boldsymbol{q}|$ is the length of the joint multigram sequence $\boldsymbol{q}$.

3.1. Model parameter estimation

Given N prompted words in the training corpus, with the V_n different mispronunciations (where $n = 1, \ldots, N$), we denote the frequency for the vth mispronunciation of the nth word by $p(\boldsymbol{\varphi}_{n,v})$, and enforce the constraint $\sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) = 1$.

Each un-aligned training sample of input-output pair serves as an observation for the JSM and can be summarized by $\mathcal{O}_{n,v} = (\boldsymbol{g}_n, \boldsymbol{\varphi}_{n,v})$ with probability $p(\boldsymbol{\varphi}_{n,v})$. Maximum Likelihood training tries to obtain the set of parameters Θ which maximizes the log-likelihood of the training samples:

$$\Theta = \arg\max \sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \log \mathcal{L}(\mathcal{O}_{n,v} | \Theta). \tag{5}$$

Note that the co-segmentation σ is hidden, and its incorporation provides the "complete-data" for EM training. As first shown in [7]:

$$\begin{aligned}
&\sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \log \mathcal{L}(\mathcal{O}_{n,v} | \Theta) \\
=&\sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \log \sum_{\sigma \in \{\sigma_{n,v}\}} \mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)}) \frac{\mathcal{L}(\sigma, \mathcal{O}_{n,v} | \Theta)}{\mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)})} \\
\geq&\sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \sum_{\sigma \in \{\sigma_{n,v}\}} \mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)}) \log \mathcal{L}(\sigma, \mathcal{O}_{n,v} | \Theta) \\
&-\sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \sum_{\sigma \in \{\sigma_{n,v}\}} \mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)}) \log \mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)}).
\end{aligned} \tag{6}$$

Eq. (6) is the auxiliary function $Q(\Theta|\Theta^{(i)})$ we wish to maximize, given $\Theta^{(i)}$ in iteration i. The "expectation" step is to evaluate $Q(\Theta|\Theta^{(i)})$, and the "maximization" step involves finding the optimal Θ that maximizes $Q(\Theta|\Theta^{(i)})$.

For a particular σ of $\mathcal{O}_{n,v}$, the application of Eq. (4) can decompose the log-likelihood $\log \mathcal{L}(\sigma, \mathcal{O}_{n,v}|\Theta)$ into the sum of a series of multigrams conditioned on the past $\mathcal{M} - 1$ history for the $\mathcal{M}$th order model:

$$\log \mathcal{L}(\sigma, \mathcal{O}_{n,v} | \Theta) = \log p(\boldsymbol{q}|\Theta) = \sum_{j=1}^{|\boldsymbol{q}|} p(q_j | h_j; \Theta), \tag{7}$$

where we introduce the symbol h to denote the sequence of preceding joint multigrams, i.e. $h_j = (q_{j-M+1}, \ldots, q_{j-1})$. The model parameters are $\{p(q|h)\}$, i.e. the probability of multigram q given the history h.

To maximize $Q(\Theta|\Theta^{(i)})$, one can formulate a Lagrangian by imposing on the auxiliary function an equality constraint that the model parameters $\{p(q|h;\Theta)\}$ sum up to one. By setting the derivate of the Lagrangian with respect to $p(q|h;\Theta)$ equal to 0, we have the re-estimation formulas:

$$\mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta) = \frac{p(\boldsymbol{q}|\Theta^{(i)})}{\sum_{\boldsymbol{q}' \in S(\mathcal{O}_{n,v})} p(\boldsymbol{q}'|\Theta^{(i)})}, \tag{8}$$

$$e(q, h|\Theta) = \sum_{n=1}^{N} \sum_{v=1}^{V_n} p(\boldsymbol{\varphi}_{n,v}) \sum_{\sigma \in \{\sigma_{n,v}\}} \mathcal{L}(\sigma | \mathcal{O}_{n,v}; \Theta^{(i)}) n_{q,h}(\boldsymbol{q}), \tag{9}$$

$$p(q|h;\Theta^{(i+1)}) = \frac{e(q, h|\Theta^{(i)})}{\sum_{q'} e(q', h|\Theta^{(i)})}, \tag{10}$$

where $n_{q,h}(\boldsymbol{q})$ is the number of occurrences of the M-gram $q_{j-M+1}, \ldots, q_j$ in $\boldsymbol{q}$. A forward-backward implementation can be used to avoid explicit search through all co-segmentations in Eq. (9) as shown in [7]. Absolute discounting with interpolation and a marginal preserving back-off distribution is applied when building higher-order models from lower-order ones, to avoid over-fitting [6].

3.2. Mispronunciation lexicon generation

The top $\mathcal{N}$ variants can be retrieved in descending posterior order by searching through a graph of $p(q|h)$ [6]:

$$p(\boldsymbol{\varphi}|\boldsymbol{g}) = \frac{\sum_{\boldsymbol{q} \in S(\boldsymbol{g}, \boldsymbol{\varphi})} p(\boldsymbol{q})}{\sum_{\sqcup \boldsymbol{q}_g = \boldsymbol{g}} p(\boldsymbol{q})}. \tag{11}$$

Since the resulting lexicon can possibly include canonical pronunciations, but we are targeting mispronunciations, we exclude the canonical pronunciations and ensure that the posteriors of the remaining entries sum to unity.

4. Experiments

4.1. Dataset

Model training and evaluation is based on the Cantonese subset of the **CH**inese **L**earners **O**f **E**nglish (**CHLOE**) corpus. This dataset includes English recordings from 50 male and 50 female native Cantonese speakers. Each speaker reads prompted text from four categories: minimal pairs, confusable words, phonemic sentences and the AESOP's fable, "The North Wind and the Sun". Each recorded utterance is transcribed by trained linguists using the ARPABET, augmented with three additional

[1] We set $L_G = L_\Phi = 1$ throughout the experiments in our paper, because these parameters yield the best result.

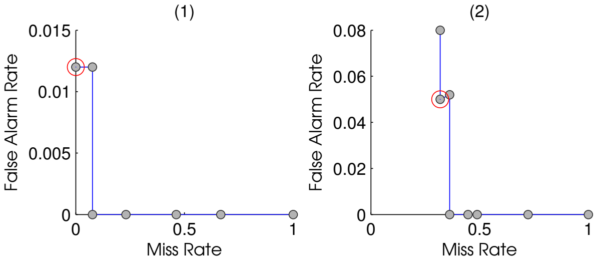

Figure 1: *Illustration of the search strategy in the FAR versus MR curve.*

symbols, namely /ax/ to denote the "schwa", /axr/ to denote the retroflexed schwa and /ix/ to denote the unstressed /ih/ according to the TIMIT convention.

We randomly split the corpus into two halves by speakers to form the training and test sets. This splitting aims to capture the commonalities of mispronunciations across speakers. It is noted that about 1449 mispronunciation patterns (about 44%) in the training set are found in the test set.

4.2. Performance metrics

Given that our objective is mispronunciation detection through modeling errors in the pronunciation lexicon, evaluation of the generated lexicon for each word can be made in terms of the *expected false alarm rate* (**FAR**) and *expected miss rate* (**MR**). The expected false alarm rate is estimated by the sum of the posteriors of mispronunciation entries which do not appear in the test set; The expected miss rate is determined by the sum of the frequencies of mispronunciations in the test set that are missing from the lexicon.

Compared with the assumption on uniform distribution with respect to the mispronunciations for each word in [5], greater penalty will be incurred if the lexicon has missed a highly frequent mispronunciation or has been too confident on a potential mispronunciation without support. The setting is also consistent with the training procedure where the frequencies of mispronunciations are taken into account as in Eq. (5). In other words, this training scheme makes common multigram patterns more representable by the model than the less common ones.

For a particular model and a particular word, a scatter chart suggesting the correlation between the two metrics can be plotted by sliding the parameter $\mathcal{N}$ - the number of top ranking mispronunciation entries by posterior. Conceivably, the curve has a distinctive saw-tooth shape and shows a negative correlation between the FAR and MR on the whole due to an intrinsic trade-off between the two measures. For the sake of comparison, we search for the "local optimum" in terms of simultaneously low FAR and MR as the overall performance indicator. Our search strategy (see Figure 1) starts from the point of 100% miss and 0% false alarm with $\mathcal{N} = 0$, and increments $\mathcal{N}$ until either: (1) MR is zero; or (2) the curve jags up and to the left in the first local minimum.

Furthermore, the false alarm rate and miss rate measured on individual words can be easily manipulated to indicate the performance on the whole set by a weighted sum of these statistics, where the weights are the frequencies of words. It is noted that the lexicon consists of all mispronunciations in the training set would give an FAR of 0.225 and an MR of 0.216 on the test set.

4.3. G2M vs. P2M

We claim in [5] that $p(\boldsymbol{\varphi}|\boldsymbol{g})$ is a more realistic formulation to generate L2 mispronunciations in prompted speech than the previous assumed $p(\boldsymbol{\varphi}|\boldsymbol{\chi})$ in the data-driven phonological rule approach, where $\boldsymbol{\chi}$ is the canonical pronunciation. The dependence of $\boldsymbol{\varphi}$ on $\boldsymbol{g}$ encapsulates both the grapheme-to-phoneme errors and L1 negative transfer effect, which is less biased than $p(\boldsymbol{\varphi}|\boldsymbol{\chi})$.

To verify the claim on top of JSM, we first train P2M joint-sequence multigram model on pairs of the canonical baseform and mispronunciation, to emulate the behavior of phonological rules. Since some of the words may have multiple canonical pronunciations, we divide the $p(\boldsymbol{\varphi}_{n,v})$ in Eq. (5) evenly among the canonical baseforms. To compare fairly with P2M, G2M models are trained on pairs of orthographic form and mispronunciation. The results are depicted in Table 1. When the order is low, the P2M out-performs the G2M, which shows that the alignment between the orthographic form and the mispronunciation transcriptions is less satisfactory than the phonetic alignment between the canonical baseform and the mispronunciation transcription. As the model order grows, the long-span dependency is captured by the $\mathcal{M}$-multigram approximation and the G2M lexicon reaches a better convergence than that of the P2M.

Another probabilistic way of explaining G2M model's better performance is that the implicit decomposition of $p(\boldsymbol{\varphi}|\boldsymbol{g})$ better fits the training data, since $p(\boldsymbol{\varphi}|\boldsymbol{g}) = \sum_{\boldsymbol{\xi}} p(\boldsymbol{\varphi}|\boldsymbol{\xi})p(\boldsymbol{\xi}|\boldsymbol{g})$ is more flexible than $p(\boldsymbol{\varphi}|\boldsymbol{\chi}) = \frac{1}{|\{\boldsymbol{\chi}_i\}|}\sum_i p(\boldsymbol{\varphi}|\boldsymbol{\chi}_i)$, where $\boldsymbol{\chi}$ may be only a subset of $\boldsymbol{\xi}$.

It is also interesting to see that the number of unseen patterns in the lexicon attains maximum when the order is low. This might suggest how L2 learners decompose a grapheme string, match a partial pronunciation locally and re-assemble to yield a pronunciation.

Besides, in the learned joint-sequence inventory, the grapheme-to-phoneme pair "R→[]" and the phoneme-to-phoneme pair "r→[]" receive the highest unigram probabilities, showing the universal 'r-deletion' phenomenon in our data.

Table 1: *Comparison of lexicons generated by P2M model with those by G2M model on different M-multigram orders. The "unseen" column shows the fraction of generated lexicon that is not seen in the training set*

	P2M			G2M		
order	FAR	MR	unseen	FAR	MR	unseen
1	0.118	0.555	158/523	0.096	0.873	44/125
2	0.078	0.328	**161**/990	0.099	0.407	**187**/809
3	0.045	0.270	76/1193	0.058	0.286	87/1115
4	0.035	0.254	35/1253	0.043	0.245	42/1283
5	0.036	**0.252**	29/1278	0.039	**0.241**	30/1295

4.4. Erroneous application of partial analogy

As implied in the previous section, the "*mispronunciations by analogy*" can be derived by matching substrings of the prompted orthographic form to substrings of known words, hypothesizing a partial pronunciation for each matched substring, and assembling the partial pronunciations. It is pointed out in [5] that there can be two cases of mispronunciation by analogy which constitute the majority of grapheme-to-phoneme errors: one is the replication of mispronounced substrings, e.g. for the same learner, the mispronounced substring of "WRATH" / *w ao th* / can recur in the mispronunciation of "WRAPPED" / *w ao p t* /; the other is the erroneous application of partial analogy, or misuse of letter-to-sound rules, e.g. mispronouncing "ANALYST" as / *ae n ax l ay s ih s* / by analogizing "ANALYZE". We devise the following two experiments to investigate the latter case.

Table 2 shows the performance of models trained on pairs of orthographic forms and canonical pronunciations for the 435 words that we use in the training set. We find that the canonical model can actually produce mispronunciations with high probabilities, e.g. "ACHING" / *ae ch ix ng* /, "DOUBT" / *d aw b t* /, "ZEALOUS" / *z iy l ax s* /, etc.

Table 2: *FAR and MR on the mispronunciation lexicon generated by canonical models.*

	CHLOE canonicals	
order	FAR	MR
1	0.038	0.937
2	0.061	0.795
3	0.100	**0.786**
4	0.104	0.787
5	0.099	0.788

We conduct another experiment by training a G2M model on all grapheme-to-phoneme pairs in the training set, i.e. including both the canonical pronunciations and mispronunciations, and we compare it with the G2M model in Section 4.3 as shown in Table 3. The canonical pronunciations plays an active role in generating additional mispronunciations, bringing down MR.

Table 3: *Comparison of G2M models trained on only mispronunciations (G2M^a) with the ones trained on both canonical pronunciations and mispronunciations (G2M^b).*

	G2M^a			G2M^b		
order	FAR	MR	unseen	FAR	MR	unseen
1	0.096	0.873	44/125	0.099	0.913	38/101
2	0.099	0.407	**187**/809	0.103	0.468	**165**/705
3	0.058	0.286	87/1115	0.065	0.277	97/1151
4	0.043	0.245	42/1283	0.045	0.243	44/1273
5	0.039	**0.241**	30/1295	0.039	**0.238**	32/1305

4.5. Recognition result

We normalize the phonetic transcriptions on TIMIT to match the phonetic code in our L2 corpus and train mono-phone HMMs as a seed and alignment model for the WSJ1 corpus. The alignment is based on a CMU dictionary which is also normalized in terms of phonetic symbols, e.g. /ah0/ and /ah1/ are converted to /ax/ and /ah/ respectively based on syllable structure analysis. Maximum Likelihood re-estimation on WSJ1 yields triphone tied-state HMMs with 16 mixtures per state. The native acoustic model is adapted on the training set of CHLOE Cantonese subset using **C**onstrained **M**aximum **L**ikelihood **L**inear **R**egression (CMLLR).

To construct the lexicon for recognition, we append not only the mispronunciations from G2M JSM, but the canonicals as well. Since the probabilities of the mispronunciation lexicon entries should add up to one, we first compute the likelihood of each entry according to Eq. (2) and Eq. (7) using the JSM model, and normalize them to keep the total posterior unity.

The statistics on the result of mispronunciation detection and diagnosis[2] using the lexicons derived by P2M, G2M^a and G2M^b are shown in Table 4. The G2M^b model yields the best performance in diagnosing mispronunciations, but is worse than G2M^a in detecting mispronunciation simply because the model tends to assign higher likelihood to the canonical pronunciations. This strong prior not only keeps the false rejection rate low but also raises the false acceptance rate.

Table 4: *Performance comparison of P2M, G2M^a and G2M^b on mispronunciation detection and diagnosis.*

	canonicals		mispronunciations		
				True Rej.	
model	True Acpt.	False Rej.	False Acpt.	Correct Diag.	Diag. Error
P2M	72.55%	27.45%	21.45%	54.38%	45.62%
G2M^a	72.58%	27.42%	**21.30%**	54.39%	45.61%
G2M^b	**74.37%**	**25.63%**	22.80%	**54.55%**	**45.45%**

5. Conclusions and Perspectives

The paper has presented a framework for generating mispronunciation lexicon used by ASR-based L2 mispronunciation detection and diagnosis. Empirical studies show the grapheme-to-mispronunciation conversion can be a better process than the previous phoneme-to-mispronunciation transfer. Statistics have confirmed the advantage of the former in low lexicon false alarm rate and low miss rate, as well as better recognition performance. More aggressive search strategy in the FAR-MR plot combined with acoustic model discriminative training [4] is worth studying to reveal the optimal operating point for diagnosis. Proper rejection mechanism can be applied to filter those un-modeled or spurious mispronunciations to complete this framework.

6. Acknowledgment

This work is affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies. It is conducted under the MSRA-CUHK joint laboratory scheme, and is partially supported by the NSFC/RGC Joint Research Scheme (project no. N_CUHK 414/09).

7. References

[1] Witt, S.M., Young, S.J., "Phone-level Pronunciation Scoring and Assessment for Interactive Language Learning", Speech Communication Vol. 30, pp. 95-108, 2000.

[2] Strik, H., Truong, K., Wet, F.D., Cucchiarini, C., "Comparing classifiers for pronunciation error detection", Interspeech 07, pp. 1837-1840, 2007.

[3] Harrison, A. M., Lo, W., Qian, X., Meng, H., "Implementation of an Extended Recognition Network for Mispronunciation Detection and Diagnosis in Computer-Assisted Pronunciation Training", 2nd ISCA Workshop on SLaTE, 2009.

[4] Qian, X., Soong, F., Meng, H., "Discriminative Acoustic Models for Mispronunciation Detection and Diagnosis and Computer Assisted Pronunciation Training", Interspeech 10, 2010.

[5] Qian, X., Meng, H., Soong, F., "Capturing L2 Segmental Mispronunciations with Joint-sequence Models in Computer-Aided Pronunciation Training (CAPT)", 7th ISCSLP, 2010.

[6] Bisani, M., Ney, H., "Joint-sequence Models for Grapheme-to-Phoneme Conversion", Speech Communication Vol. 50, pp. 434-451, 2008.

[7] Sabine, D., Frederic, B., "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", International Conference on Acoustics, Speech and Signal Processing, 1995.

[2] We do not tune the weights of acoustic likelihood and dictionary probability on a development set.