Mispronunciation Detection and Diagnosis in L2 English Speech Using Multidistribution Deep Neural Networks

Kun Li, Xiaojun Qian, and Helen Meng, *Fellow, IEEE*

***Abstract*—This paper investigates the use of multidistribution deep neural networks (DNNs) for mispronunciation detection and diagnosis (MDD), to circumvent the difficulties encountered in an existing approach based on extended recognition networks (ERNs). The ERNs leverage existing automatic speech recognition technology by constraining the search space via including the likely phonetic error patterns of the target words in addition to the canonical transcriptions. MDDs are achieved by comparing the recognized transcriptions with the canonical ones. Although this approach performs reasonably well, it has the following issues: 1) Learning the error patterns of the target words to generate the ERNs remains a challenging task. Phones or phone errors missing from the ERNs cannot be recognized even if we have well-trained acoustic models; and 2) acoustic models and phonological rules are trained independently, and hence, contextual information is lost. To address these issues, we propose an acoustic-graphemic-phonemic model (AGPM) using a multidistribution DNN, whose input features include acoustic features, as well as corresponding graphemes and canonical transcriptions (encoded as binary vectors). The AGPM can implicitly model both grapheme-to-likely-pronunciation and phoneme-to-likely-pronunciation conversions, which are integrated into acoustic modeling. With the AGPM, we develop a unified MDD framework, which works much like free-phone recognition. Experiments show that our method achieves a phone error rate (PER) of 11.1%. The false rejection rate (FRR), false acceptance rate (FAR), and diagnostic error rate (DER) for MDD are 4.6%, 30.5%, and 13.5%, respectively. It outperforms the ERN approach using DNNs as acoustic models, whose PER, FRR, FAR, and DER are 16.8%, 11.0%, 43.6%, and 32.3%, respectively.**

***Index Terms*—Deep neural networks, L2 English speech, mispronunciation detection, mispronunciation diagnosis, speech recognition.**

I. Introduction

COMPUTER-AIDED pronunciation training (CAPT) technologies enable self-directed language learning with round-the-clock accessibility and individualized feedback. They can supplement the teachers' instructions and help meet the demand of a growing population of learners in face of a shortage of qualified teachers. CAPT technologies focus on mispronunciation detection and diagnosis (MDD)—the former decides whether the learner's articulation is correct or incorrect, while the latter identifies the specific error(s) to generate corrective feedback and facilitate learning. In some sense, MDD are more challenging than direct automatic speech recognition (ASR), which aims to transcribe speech input regardless of the pronunciation accuracy. The performance of ASR also needs marked improvements in order to adequately support MDD in CAPT. This is because the acoustic models need to discriminate the canonical phonetic pronunciations from the non-native deviants, some of which may be subtle differences.

MDD can be implemented at segmental and supra-segmental levels [1]. The segmental level involves phones and words; while the suprasegmental level includes lexical stress [2]–[8], pitch accent [3], [5], [8]–[11] intonation [12], [13], rhythm, etc. In this work, we only consider the segmental level (phone).

Previous studies that examine the speech uttered by second-language (L2) learners stated three kinds of causes producing phone-level mispronunciations [14], [15]: (1) *Language transfer*—learners tend to replace a phone in the target language that is absent from their mother tongue with another phone that is present. For example, "north" $/n\ ao\ r\ th/$ is usually realized as $/n\ ao\ f/$ by native Cantonese speakers learning English. (2) *Incorrect letter-to-sound conversion*—for unfamiliar words, L2 learners may pronounce these by guessing. For example, "hypnotic" $/hh\ ih\ p\ n\ ao\ t\ ih\ k/$ mispronounced as $/hh\ ay\ p\ n\ ao\ t\ ih\ k/$. (3) *Misreading the text prompts*, e.g., "when" misread as "then".

Language transfer effects contribute to mispronunciations in non-native productions [16], and many studies (e.g., [17]–[21]) focus on this type of phonetic error. In this work, we will try to handle all the three kinds of errors above in a unified approach.

II. Previous Work to Achieve Phone-Level MDD

CAPT systems leverage the advancements in speech and language technologies to achieve automatic mispronunciation detection and diagnoses—the two processes that are most conducive to learning. Previous work have predominantly focused on mispronunciation detection, which is the task that precedes diagnoses. Related studies covers a diversity of primary (L1) and secondary (L2) languages. Specific efforts that support Chinese learners of English include [22]–[29]. Previous work to achieve MDD can be grouped into several categories:

Manuscript received November 15, 2015; revised March 29, 2016, July 9, 2016, and September 11, 2016; accepted September 29, 2016. Date of publication November 22, 2016; date of current version December 15, 2016. The work was supported in part by a grant from the HKSAR Government Research Grants Council General Research Fund (project number 14207315). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tatsuya Kawahara.

The authors are with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong (e-mail: kli@se.cuhk.edu.hk; xjqian@se.cuhk.edu.hk; hmmeng@se.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2016.2621675

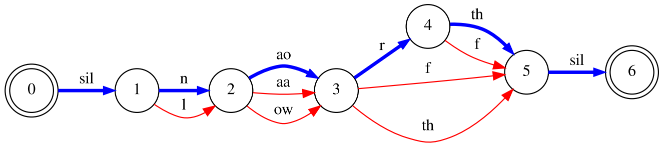


Fig. 1. Illustration of the ERN for the word "north" [20]. The canonical path (the topmost one) is highlighted in bold blue.

A. Approaches Based on Pronunciation Scoring

Different types of confidence measures have been used as pronunciation scores, e.g., phone durations [30]–[32], loudness [25], likelihoods [30], [31], [33], [34], likelihood ratios [35]–[37], phone posterior probabilities [31], [32], [34] and their combinations [31], [32]. Based on the likelihoods, Witt and Young [18] proposed the well-known Goodness of Pronunciation (GOP). The scores can be obtained from the acoustic models trained on native-like productions or an additional set of acoustic models trained on incorrect non-native productions [32]. Mispronunciation detection is achieved by thresholding pronunciation scores, often with phone-dependent thresholds. Recent efforts use multiple log-likelihood ratios (LLRs) for each target phone, where the ratios correspond to multiple acoustically similar phones (i.e., variants) found in mispronunciations. For example, van Doremalen *et al.* [38] extended GOP to weighted GOP that combines multiple LLRs by logistic regression.

Lee and Glass [27] developed a comparison-based framework to detect the word-level mispronunciations. Dynamic time warping is run between the student's utterance and the teacher's utterance and the alignment is based on the similarity of Mel-frequency cepstral coefficients (MFCCs), Gaussian posteriorgrams or DNN posteriorgrams [28]. A pronunciation scoring method [39] is also proposed using this approach. This method works well on detecting mispronunciations when reference speech is given. However, it does not diagnose the detected mispronunciations.

The challenge is that the discriminating correct, native-like productions from incorrect, non-native ones based on solely phone-level scoring algorithms can be difficult. Longer segments of speech serves better for human judgment and can offer a good indication to overall proficiency, while with shorter phone segments even human judgment can be less consistent [40], let along automatic scoring.

B. Approaches Based on Forced Alignment Using Extended Recognition Networks (ERNs)

Ronen *et al.* [41] first built the mispronunciation networks that include two sets of phone-based acoustic models and allow transitions between them. One set is trained on native speech and the other on non-native. After forced alignment, the path with the highest probability can be identified. If this path includes non-native models, then the corresponding segments are regarded as mispronunciations. Subsequent to this work, ERNs were proposed to cover not only the canonical transcriptions but some likely error patterns as well [17]–[21], [42], [43]. An example from Harrison *et al.* [20] is shown in Fig. 1.

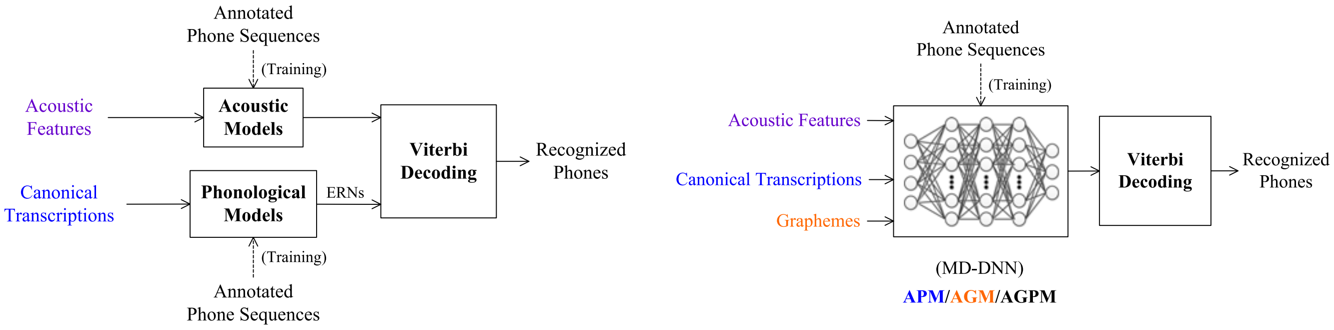
To build ERNs, one efficient way is to make use of context-dependent phonological rules, which can be hand-crafted [14], [19], [20] or derived with data-driven approaches [16]. The phonological rules perform context-dependent transductions from canonical phonetic transcriptions. In addition, Qian *et al.* [15], [44] proposed to use a joint-sequence multi-gram model (JSM) [45] to generate the possible error patterns caused by *incorrect letter-to-sound conversion*. The JSM is trained on pairs of grapheme of target words and the corresponding annotated phone sequences of learners' speech. The former way is a kind of phoneme-to-likely-pronunciation (P2LP) conversion; while the latter is grapheme-to-likely-pronunciation (G2LP) conversion. Experiments in [44] shows that the G2LP JSM may cover more mispronunciation variants and achieve better recognition performance than the data-driven P2LP phonological rules. The diagram of the approach using ERNs generated by P2LP phonological models is shown in Fig. 2(a).

This ERN approach enables mispronunciation diagnosis in addition to mispronunciation detection during recognition (see Fig. 3). However, they have the following shortcomings: (a) It is difficult to guarantee a high coverage of possible mispronunciation, due to unanticipated candidates in manual rule authoring or unseen observations in data-driven approaches. Phones missing from ERNs cannot be recognized even if we have well-trained acoustic models. (b) There is a trade-off between coverage and precision. Performance would decrease greatly if too many possible mispronunciations are included. Free-phone recognition is the extreme condition where all possible alternative phones are considered (and hence all possible mispronunciations are covered). (c) ERNs are usually generated by the P2LP phonological rules or the G2LP JSM, whereas acoustic models are trained independently of the phonological context. Hence, valuable information may be lost.

C. Approaches Based on Acoustic Representations and Models

There has also been work that explores alternatives to standard acoustic representations (e.g., MFCCs) or acoustic models (e.g., hidden Markov models). Truong *et al.* [46] selected a small set of discriminative features and developed a specific linear discriminant analysis (LDA) classifier for detecting each pronunciation error of several Dutch phones. Tepperman and Narayanan [47] investigated articulatory features with reference to a phone-to-articulator mapping [48]. Richardson *et al.* [49] adopted hidden-articulator Markov models to generate scores and feature vectors. Recent work use posteriorgrams to represent the L2 English acoustic-phonetic space. Posteriorgrams are vectors of class-based posterior probabilities previously used in unsupervised keyword detection [50], [51] and can capture acoustic-phonetic characteristics in a discriminative and robust manner [52]. Posteriorgrams used for mispronunciation detection include those generated by Gaussian mixture models (GMMs) [27], deep neural networks (DNNs) [28] and also multi-layer perceptrons [42].

In addition, deep learning techniques [53], [54] have recently been shown to be highly effective in many pattern recognition



(a) ERN Approach (b) Proposed Approach

Fig. 2. Diagrams of the ERN and APM/AGM/AGPM approaches. Canonical transcriptions are extracted from dictionaries according to the words prompted to L2 learners. Annotated phone sequences are only used in the training stage. The ERNs in (a) are generated by phoneme-to-likely-pronunciation (P2LP) phonological models. Besides acoustic features, the APM in (b) makes use of canonical transcriptions, whereas the AGM leverage graphemes. The input features of the AGPM include acoustic, graphemic and phonemic features.

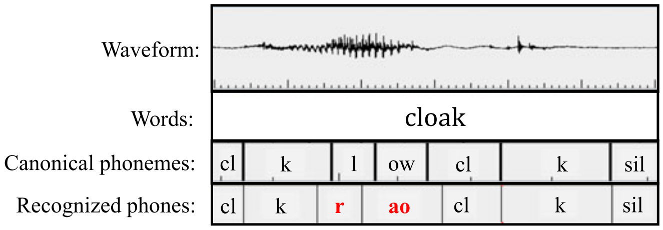


Fig. 3. A segment of L2 English speech with aligned canonical phonemic sequence and recognized phone sequence. By further aligning the two sequences, the mispronunciations produced by the L2 speaker can be detected and diagnosed, i.e., /l/ and /ow/ are mispronounced as /r/ and /ao/ respectively. These mispronunciations are marked in bold red.

tasks. In the field of ASR, DNNs are used to replace GMMs as part of acoustic models and achieved significant performance improvements [55]–[57]. Many derivatives of DNNs, such as deep convolutional neural networks [58], [59] and deep recurrent neural networks [60], also achieved impressive performance improvements. Their phone recognition error rates over the TIMIT corpus are below 20% [55], [59]–[61]. Qian *et al.* [62] first applied deep learning techniques to mispronunciation detection and diagnosis. DNNs are used to replace GMMs to model the phone-state posteriors in Hidden Markov Models. Hu *et al.* [63] refined acoustic models by DNN training with native American English speech for pronunciation assessment based on GOP scores. This is followed by the use of a neural network-based logistic regression classifier to achieve improved mispronunciation detection [64], [65]. As mentioned above, Lee *et al.* [28] also used a DNN to improve posteriorgrams in mispronunciation detection. Most existing approaches simply adopted DNNs to replace GMMs as part of acoustic models. However, there is a need for significant performance improvement for mispronunciation detection and diagnosis.

III. Introduction to Multi-Distribution DNNs (MD-DNNs)

In real applications, input features may have different kinds of distributions, e.g., some features maybe binary and the others maybe Gaussian. To incorporate these features, Kang *et al.* [66], [67] proposed an MD-DNN for speech synthesis, which was also applied to lexical stress detection [7]. Similar to traditional DNNs, they are also constructed by stacking up multiple Restricted Boltzmann Machines (RBMs) from bottom up. This involves running a layer-by-layer unsupervised pre-training algorithm [53], [54], followed by fine-tuning the pre-trained network using the back-propagation algorithm [68]. Excluding the bottom RBM, all the other ones are traditional Bernoulli RBM, whose hidden and visible units are all binary. The bottom RBM is a type of mixed Gaussian-Bernoulli RBM, whose hidden units are binary while visible units maybe Gaussian or binary.

A. Bernoulli RBM (B-RBM)

The energy of the joint configuration of visible and hidden vectors $(\mathbf{v}, \mathbf{h})$ is given as:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\Theta}) = -\mathbf{h}^{\mathrm{T}}\mathbf{W}\mathbf{v} - \mathbf{a}^{\mathrm{T}}\mathbf{h} - \mathbf{b}^{\mathrm{T}}\mathbf{v} \tag{1}$$

where $\boldsymbol{\Theta} = (\mathbf{W}, \mathbf{a}, \mathbf{b})$ is the set of parameters of an RBM and $\boldsymbol{\Theta}$ will be omitted for clarity hereafter. $\mathbf{W}$ is the matrix of visible/hidden connection weights, $\mathbf{a}$ is the hidden unit bias and $\mathbf{b}$ is the visible unit bias.

The probability is given in terms of the energy:

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{\sum_{\tilde{\mathbf{v}}} \sum_{\tilde{\mathbf{h}}} e^{-E(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})}} \tag{2}$$

Since there is no connection within a layer and the units are all binary, we obtain [57]:

$$p(h_j = 1 \mid \mathbf{v}) = \sigma\left(\sum_i w_{ij} v_i + a_j\right) \tag{3a}$$

$$p(v_i = 1 \mid \mathbf{h}) = \sigma\left(\sum_j w_{ij} h_j + b_i\right) \tag{3b}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$.

B. Mixed Gaussian-Bernoulli RBM (GB-RBM)

The energy of the joint configuration of the visible and hidden vectors is given as [66]:

$$E(\mathbf{v}^g, \mathbf{v}^b, \mathbf{h}) = -\mathbf{h}^{\mathrm{T}}\mathbf{W}^g\mathbf{v}^g + \frac{1}{2}(\mathbf{v}^g - \boldsymbol{\mu})^{\mathrm{T}}(\mathbf{v}^g - \boldsymbol{\mu}) - \mathbf{h}^{\mathrm{T}}\mathbf{W}^b\mathbf{v}^b - \mathbf{b}^{\mathrm{T}}\mathbf{v}^b - \mathbf{a}^{\mathrm{T}}\mathbf{h} \tag{4}$$

where $\mathbf{v}^b$, $\mathbf{v}^g$ are the Bernoulli units and linear units with Gaussian noise in the visible layer, $\mathbf{W}^b$ and $\mathbf{W}^g$ are the respective weight matrices, $\boldsymbol{\mu}$ is the mean of $\mathbf{v}^g$, $\mathbf{a}$ and $\mathbf{b}$ are the bias terms of $\mathbf{h}$ and $\mathbf{v}^b$.

The following conditional probabilities can be derived for pre-training and fine-tuning a GB-RBM [66]:

$$p(h_j \mid \mathbf{v}^g, \mathbf{v}^b) = \sigma\left(\sum_i w_{ij}^g v_i^g + \sum_i w_{ij}^b v_i^b + a_j\right) \tag{5a}$$

$$p(v_i^b = 1 \mid \mathbf{h}) = \sigma\left(\sum_j w_{ij}^b h_j + b_i\right) \tag{5b}$$

$$p(v_i^g \mid \mathbf{h}) = \mathcal{N}\left(\mu_i + \sum_j w_{ij} h_j, 1\right) \tag{5c}$$

where $\mathcal{N}(\mu_i + \sum_j w_{ij} h_j, 1)$ denotes a Gaussian with mean $(\mu_i + \sum_j w_{ij} h_j)$ and variance 1.

Comparing the conditional probabilities of GB-RBM given in (5a)–(5c) with those of B-RBM given in (3a) & (3b), we observe that the main difference lies in (5c). If there are only binary units in the visible layer of GB-RBM, (5a) & (5b) are equivalent to (3a) & (3b), i.e., B-RBM is a special type of GB-RBM. Another type of RBM, whose hidden units are binary while visible units are all linear units with Gaussian noise, is also a special type of GB-RBM and widely used in ASR [55]–[57].

IV. Motivation and Overall Approach

As introduced in Section II-B, previous work [17]–[20] trained acoustic models independently of P2LP phonological models (see Fig. 2(a)), and hence contextual information is lost. In order to integrate these two types of models, we first propose the Acoustic-Phonemic Model (APM)[1], as shown in Fig. 2(b). In CAPT systems, the prompts for L2 learners to utter are usually carefully designed and thus the canonical transcriptions can be known in advance. To incorporate acoustic features (assumed to have Gaussian distribution) and corresponding canonical transcriptions (encoded as binary vectors), an MD-DNN is adopted to infer the pronunciations of L2 learners as accurately as possible. We believe that this MD-DNN can implicitly learn the phonological transductions from the canonical and annotated phone sequences, and the transductions can augment the acoustic features.

Similar to the APM which assumes that mispronunciations are realized as phonetic transductions, we propose an Acoustic-Graphemic Model (AGM) which implicitly models the pronunciations from the surface forms of the target words—called graphemes. The AGM is also an MD-DNN that makes use of acoustic and graphemic features. The APM and AGM can be further combined into an Acoustic-Graphemic-Phonemic Model (AGPM) whose input features include acoustic, graphemic and phonemic features. We conjecture that graphemes and phonemes may complement each other, so that performance maybe improved further.

In this paper, we develop a unified mispronunciation detection and diagnosis framework which works much like free-phone recognition. The only difference between them is that the our approach makes use of prompted texts while the traditional free-phone recognition does not. Using the prompts, we propose the APM/AGM/AGPM to take the place of conventional acoustic models. When the recognized phones differ from the canonical transcriptions (obtained from the text prompts presented to the speaker), mispronunciation detection and diagnosis are achieved by aligning the two, as illustrated in Fig. 3.

For the sake of clarity and comparison, we first realize traditional free-phone recognition. A monophone acoustic model and a phone-state transition model are built, both of which are DNNs. Then the APM incorporating acoustic and phonemic features is built. The APM is further extended to the AGPM which integrates graphemic features in addition to acoustic and phonemic features. The structure of our paper is designed as follows: Section V describes the free-phone recognition for L2 English speech; Section VI introduces our approach using the APM; Section VII describes the AGPM; Sections VIII and IX presents the experimental setups and results respectively. Finally, conclusions are given in Section XI.

V. Free-Phone Recognition for L2 English Speech (Baseline)

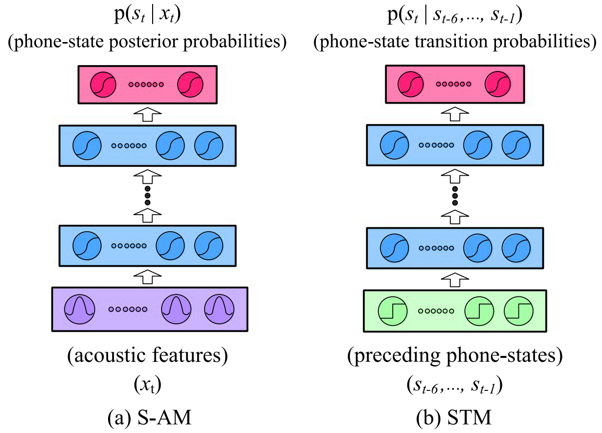
To realize free-phone recognition for L2 English, a State-level Acoustic Model (S-AM) and a State Transition Model (STM) are built [70], both of which use DNNs.

A. State-level Acoustic Model (S-AM)

The speech is sampled at 16 kHz. To compensate for the high-frequency part of speech signal, a pre-emphasis filter is applied to the speech, whose transfer function is $1 - 0.97z^{-1}$. Then Fast Fourier Transform is performed in a 25-ms Hamming window with a 10-ms frame shift. Finally, a set of 13 MFCC features are computed per 25-ms frame. Cepstral mean normalization is done for each utterance and the features are further scaled to have zero mean and unit variance over the whole corpus.

The diagram of our acoustic model is shown in Fig. 4(a). In our experiments, we use 21 frames (1 current, 10 before and 10 after) of MFCCs as the input features x_t, thus there are 273 linear units with Gaussian noise in the bottom of the DNN. For the top layer, there are 90 units generating the posterior probabilities $p(s_t \mid x_t)$, where s_t denotes the 90-phone-state vector of the t^{th} frame. Between the bottom and top layers, there are four hidden layers and each has 512 units. Note that the number of hidden layers and the size of each layer are determined by

[1] The APM was published in our conference paper [69] and will be reintroduced here with slight modification, as well as with more detailed and updated analysis.


p(s_t | x_t)
(phone-state posterior probabilities)
p(s_t | s_{t-6},..., s_{t-1})
(phone-state transition probabilities)
(acoustic features)
(x_t)
(preceding phone-states)
(s_{t-6},..., s_{t-1})

(a) S-AM (b) STM

Fig. 4. Diagrams illustrating the proposed State-level Acoustic Model (S-AM) and State Transition Model (STM).

experiments with different configurations, which are similar to those described in Section IX-C.

To obtain the 90 phone states, we first divide each annotated phone equally into three parts to train the S-AM. Based on the 48-phone set following [71] and 3 states per phone, there are a total of 144 phone states in the output layer of the DNN. With this trained S-AM, we perform forced alignment over the entire corpus based on the annotated phone sequences and merge the state with the lowest occurrence into their neighboring states of the same phones. With the new phonetic boundaries, we re-train the S-AM. These two steps are repeated until the occurrence rate of each phone state is above a typical threshold of 0.5%. Finally, we have a set of 90 phone-states in this work.

B. State Transition Model (STM)

To generate the probabilities of phone-state transition, we build a 7-gram STM, whose diagram is shown in Fig. 4(b). Since the elements of the 90-phone-state set can be represented by 7 bits, we use 42 binary input units to indicate the previous 6 phone states $(s_{t-6}, \cdots, s_{t-1})$. Above the bottom layer, there are four hidden layers and each has 256 units. This configuration is also determined by experimentation. For the top softmax output layer, there are 90 units generating the phone-state transition probabilities $p(s_t \mid s_{t-6}, \cdots, s_{t-1})$.

C. Phone Recognition

We determine the phone-state sequence with the highest posterior probability using Viterbi decoding and treat it as the recognized phone-state sequence, as given in (6):

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} p(\boldsymbol{s} \mid \mathbf{x}) \tag{6}$$

where $\mathbf{x}$ is the sequence of acoustic feature vectors and $\boldsymbol{s}$ denotes a possible phone-state sequence.

The posterior probability of $\boldsymbol{s}$ given $\mathbf{x}$ is:

$$\begin{aligned} p(\mathbf{s} \mid \mathbf{x}) &= p(s_1|\mathbf{x})\, p(s_2|s_1, \mathbf{x}) \cdots p(s_t|s_1, \cdots, s_{t-1}, \mathbf{x}) \cdots \\ &\approx p(s_1|x_1)p(s_2|s_1, x_2) \cdots p(s_t|s_{t-6}, \cdots, s_{t-1}, x_t) \cdots \end{aligned} \tag{7}$$

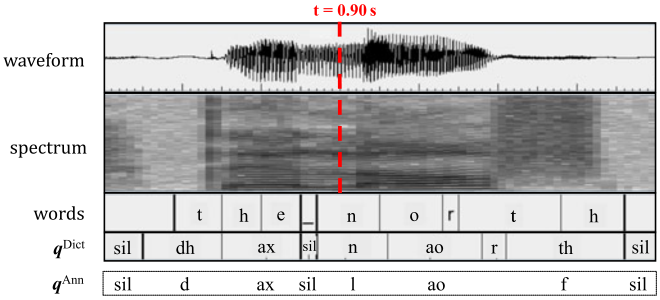


Fig. 5. An example of L2 English speech aligned with canonical phonemic sequence $\boldsymbol{q}^{\text{Dict}}$. The annotated phone sequence $\boldsymbol{q}^{\text{Ann}}$ is also given.

where x_t and s_t are the acoustic feature vector and phone state at the t^{th} frame, respectively. Note that we use a 7-gram STM and x_t has a contextual window of (10 + 1 + 10) frames.

Applying Bayes' Theorem, we have:

$$\begin{aligned} p(s_t|s_{t-6}, \cdots, s_{t-1}, x_t) &= \frac{p(s_t)\, p(s_{t-6}, \cdots, s_{t-1}, x_t|s_t)}{p(s_{t-6}, \cdots, s_{t-1}, x_t)} \\ &\approx \frac{p(s_t)p(s_{t-6}, \cdots, s_{t-1}|s_t)p(x_t|s_t)}{p(s_{t-6}, \cdots, s_{t-1})p(x_t)} \\ &= p(s_t|s_{t-6}, \cdots, s_{t-1}) \frac{p(s_t|x_t)}{p(s_t)} \end{aligned} \tag{8}$$

From (7) and (8), we have:

$$\begin{aligned} p(\mathbf{s} \mid \mathbf{x}) \;\approx\; & p(s_1|x_1)\, p(s_2|s_1) \frac{p(s_2|x_2)}{p(s_2)} \cdots \\ & p(s_t|s_{t-6}, \cdots, s_{t-1}) \frac{p(s_t|x_t)}{p(s_t)} \cdots \end{aligned} \tag{9}$$

where $p(s_t|x_t)$ is the phone-state posterior probability from the S-AM, $p(s_t|s_{t-6}, \cdots, s_{t-1})$ is the phone-state transition probability from the STM and $p(s_t)$ is the phone-state prior probability estimated from training data.

VI. Acoustic-Phonemic Model (APM)

In this section, we describe the APM which uses an MD-DNN to incorporate acoustic and phonemic features. We first align the learners' speech with the canonical phone sequences, and then use the APM to calculate the phone-state posterior probabilities. Finally we introduce the Viterbi decoding for the APM, which is similar to the one for free-phone recognition introduced in Section V-C.

A. Forced Alignment of Canonical Phonemic Transcriptions

In order to get each frame's corresponding expected phone and the contextual phones, we use the S-AM described in last section to align the speech with the canonical phonemic sequence $\boldsymbol{q}^{\text{Dict}}$, which is derived from dictionaries according to the words prompted to readers. Fig. 5 presents an example of aligning $\boldsymbol{q}^{\text{Dict}}$ with L2 English speech uttered by a native Cantonese learner, who followed the prompt from the CAPT system to produce the words "the north".

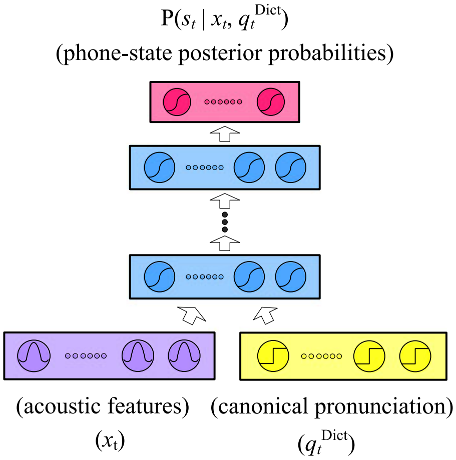

$P(s_t \mid x_t, q_t^{\text{Dict}})$
(phone-state posterior probabilities)
(acoustic features)
(x_t)
(canonical pronunciation)
(q_t^{Dict})

Fig. 6. Diagram illustrating the proposed Acoustic-Phonemic Model (APM).

Note that the phone-level transcriptions from the annotator are $/sil\ d\ ax\ l\ ao\ f\ sil/$. That is to say, there are several mispronunciations produced by the L2 learner, e.g., $/dh/$, $/n/$ and $/th/$ are mispronounced as $/d/$, $/l/$ and $/f/$, respectively. In addition, the learner failed to pronounce the $/r/$, which is acceptable in British English but treated as a mispronunciation in this work.

B. Implementation of APM Using MD-DNN

Similar to the S-AM, we use 273 MFCC features. From the forced alignment of canonical transcriptions, we obtain each frame's expected canonical phone. In this work, we use 7 canonical phones (3 before, 1 current and 3 after) around the current frame. For the example in Fig. 5, the 7 canonical phones (q_t^{Dict}) of the frame $t = 0.90$ s are $/dh\ ax\ sil\ \boldsymbol{n}\ ao\ r\ th/$, respectively. Note that we adopt the 48-phone set following [71] and each phone is encoded with 6 bits.

The diagram of our APM is shown in Fig. 6, which is an MD-DNN [7], [66]. In the bottom of the DNN, there are 273 linear units with Gaussian noise for the acoustic features x_t and 42 binary units for the canonical phone pronunciation q_t^{Dict}. The other layers are similar to those in Fig. 4(a).

Given each frame's canonical phone pronunciation q_t^{Dict} and annotated phone state s_t, the APM can implicitly learn the phonological transductions in the training stage. Taking Fig. 5 as an example, the APM may implicitly learn the transductions given in (10), if there are sufficient similar samples from training data.

$$/dh\ \ V/ \ \rightarrow\ /d\ \ V/ \tag{10a}$$

$$/n\ \ V/ \ \rightarrow\ /l\ \ V/ \tag{10b}$$

$$/V\ \ r/ \ \rightarrow\ /V\ \ \phi/ \tag{10c}$$

$$/th/ \ \rightarrow\ /f/ \tag{10d}$$

where V denotes a vowel and "ϕ" a deletion. With these learned phonological transductions, it will be much easier to predict the phone state s_t from the acoustic features x_t and the expected canonical phone pronunciation q_t^{Dict}.

C. Phone Recognition

Using the APM, the objective function in Viterbi decoding is changed from (6) to (11):

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} p(\boldsymbol{s} \mid \mathbf{x}, \boldsymbol{q}^{\mathbf{Dict}}) \tag{11}$$

where $\boldsymbol{q}^{\mathbf{Dict}}$ is the canonical phonemic sequence.

Similar to (9), we have:

$$p(\boldsymbol{s} \mid \mathbf{x}, \boldsymbol{q}^{\mathbf{Dict}}) \approx p(s_1|x_1, q_1^{\text{Dict}})p(s_2|s_1)\frac{p(s_2|x_2, q_2^{\text{Dict}})}{p(s_2)}\cdots$$
$$p(s_t|s_{t-6}, \cdots, s_{t-1})\frac{p(s_t|x_t, q_t^{\text{Dict}})}{p(s_t)}\cdots \tag{12}$$

where q_t^{Dict} is the corresponding canonical transcription with a contextual window of (3 + 1 + 3) phonemes at the t^{th} frame, and $p(s_t \mid x_t, q_t^{\text{Dict}})$ is the phone-state posterior probability computed from the APM.

VII. Acoustic-Graphemic-Phonemic Model (AGPM)

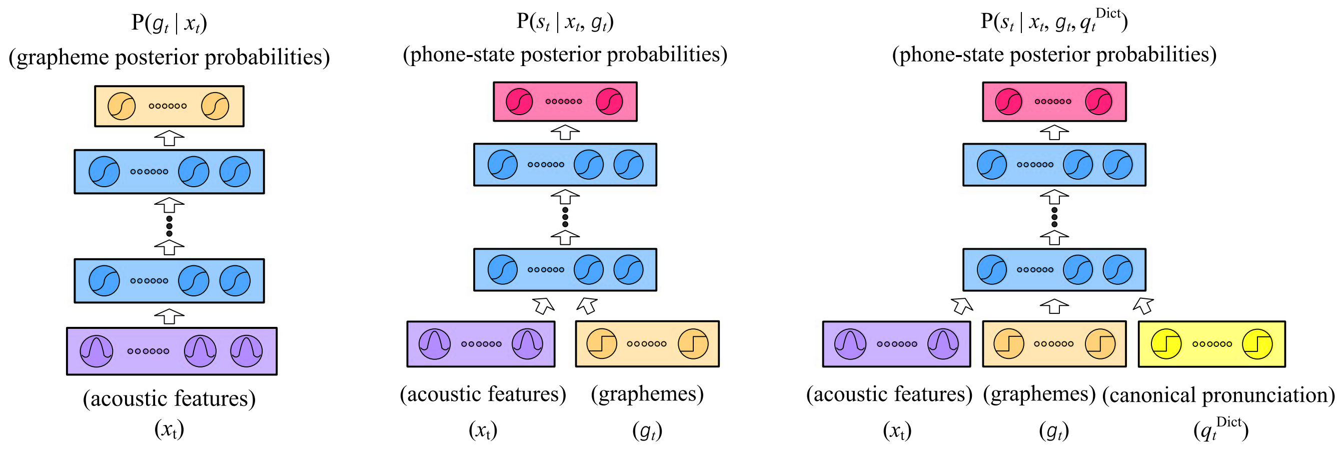
The S-AM recognizes the speech produced by L2 learners and only relies on acoustic features. The APM implicitly models the P2LP conversion and recognizes the speech from not only acoustic features but also corresponding canonical transcriptions. In this section, we will propose an Acoustic-Graphemic-Phonemic Model (AGPM) which aims to implicitly model *both* P2LP and G2LP conversions.

We will start by discussing the relationship between phonemes and graphemes in English words, then propose a Grapheme-level Acoustic Model (G-AM) which is used to align the speech with the graphemes of the prompted words. Making use of the aligned graphemes, we develop an Acoustic-Graphemic Model (AGM), which will be further extended to the AGPM.

A. Correlation Between Phonemes and Graphemes

Except for some loanwords, the phonemes and graphemes in a word have high correspondences with one another. Hence, both grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) conversions are somewhat regular. Automatic G2P (or letter-to-sound) conversion is usually developed to generate appropriate pronunciations of out-of-vocabulary (OOV) words for speech synthesis and recognition. G2P conversion can be achieved by rules [72], data-driven techniques [45], [73]–[77] or their combination [78]. In particular, G2P conversion can also be achieved by neural networks [79]–[82].

In contrast to G2P conversion which aims to generate proper pronunciations, the grapheme-to-likely-pronunciation (G2LP) conversion in MDD is designed to generate likely error patterns uttered by L2 learners as well as canonical transcriptions. In [15], [44], Qian *et al.* *explicitly* modeled the G2LP conversion using the JSM (see Section II-B), whose results are used to build the ERNs for MDD. In this section, we will propose the AGM to *implicitly* model the G2LP conversion, which is integrated into acoustic modeling for L2 English speech.


P(g_t | x_t)
(grapheme posterior probabilities)
(acoustic features)
(x_t)
P(s_t | x_t, g_t)
(phone-state posterior probabilities)
(acoustic features)
(graphemes)
(x_t)
(g_t)
P(s_t | x_t, g_t, q_t^{Dict})
(phone-state posterior probabilities)
(acoustic features)
(graphemes)
(canonical pronunciation)
(x_t)
(g_t)
(q_t^{Dict})

(a) G-AM (b) AGM (c) AGPM

Fig. 7. Diagrams illustrating the proposed Grapheme-level Acoustic Model (G-AM), Acoustic-Graphemic Model (AGM) and Acoustic-Graphemic-Phonemic Model (AGPM).

B. Grapheme-Level Acoustic Model (G-AM)

The G-AM is used to align the speech uttered by L2 learners with the graphemes of the prompted words, whose results will be used in building the AGM and AGPM. The G-AM is illustrated in Fig. 7(a). Similar to the S-AM (in Fig. 4(a)), there are also four hidden layers and one bottom layer with 273 linear units with Gaussian noise. The main difference lies in the output layer which has 28 units generating the posterior probabilities of the 28 graphemes, including the 26 letters in the English alphabet and two units representing the word boundary and the apostrophe respectively.

In order to train this G-AM, we need to know the time boundaries of graphemes, which is not provided for most corpora. To obtain their boundaries, we may make use of the word boundaries, which are usually given, or can be derived from the phone boundaries. We first divide the graphemes equally within their words to train the G-AM. Then we run forced alignment and re-train the G-AM for 3–5 iterations.

C. Acoustic-Graphemic Model (AGM)

Both APM and AGM aim to incorporate extra information regarding the generation of likely pronunciations in acoustic modeling. The APM implicitly models the P2LP conversion, whereas the AGM handles the G2LP conversion. Fig. 7(b) illustrates the diagram of our proposed AGM, which is similar to that of the APM (see Fig. 6). Their main difference lies in the fact that the AGM uses corresponding graphemes instead of canonical phonemes as part of its input features. In this work, we use 7 graphemes (3 before, 1 current, 3 after) around the current frame and each grapheme is represented by 5 bits. Take the example of the speech in Fig. 5, the 7 graphemes (g_t) corresponding to the frame t = 0.90 s are $/h\ e\ \_\ \boldsymbol{n}\ o\ r\ t/$, where "_" stands for a word boundary. Its objective function is slightly changed from (11) to (13):

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} p(\boldsymbol{s} \,|\, \mathbf{x}, \boldsymbol{g}) \tag{13}$$

where $\boldsymbol{g}$ is the grapheme sequence extracted from the prompted words, other symbols are the same as those in (6).

D. Implementation of AGPM Using MD-DNN

The APM and AGM only focus on modeling the P2LP and G2LP conversions, respectively. In order to combine these two models, we propose the AGPM whose diagram is shown in Fig. 7(c). In the bottom layer, there are 273 linear units with Gaussian noise for MFCC features (21 frames and each frame is represented by 13 MFCCs) as well as 77 binary units standing for 7 graphemes (each grapheme is described by 5 bits) and 7 canonical phonemes (each phoneme is encoded with 6 bits). Above the bottom layer, there are four hidden layers and one output layer, which are the same to the APM and AGM.

This AGPM makes use of acoustic features as well as corresponding graphemes and canonical phonemes as its input features. In this way, we attempt to implicitly model error patterns due to phonological transduction or letter-to-sound conversion, which is integrated into acoustic modeling. This approach also works much like free-phone recognition by replacing the S-AM with AGPM. Its objective function is further changed from (13) to (14):

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} p(\boldsymbol{s} \,|\, \mathbf{x}, \boldsymbol{g}, \boldsymbol{q}^{\mathbf{Dict}}) \tag{14}$$

Table I illustrates some examples of likely pronunciations, some of which can be implicitly modeled by the APM and some by the AGM. All of the pronunciations, including the correct and incorrect ones, can be modeled by the AGPM.

For the words with regular pronunciations, their canonical pronunciations can be easily predicted by their graphemes, e.g., "north" in Table I. These words' likely pronunciations can be implicitly modeled by the APM, AGM and AGPM. For the words with irregular pronunciations (e.g., "quayside" and "thyme"), some of their likely pronunciations are due to language transfer and can be modeled by the APM; while some of their error patterns are caused by letter-to-sound conversion and can be

TABLE I
EXAMPLES OF WORDS WITH LIKELY PRONUNCIATIONS MODELED BY APM, AGM AND AGPM

word	pronunciation	APM	AGM	AGPM
north	***/n ao r th/***	✓	✓	✓
	/n ao r f/	✓	✓	✓
	/n ao s/	✓	✓	✓
	/l ao f/	✓	✓	✓
quayside	***/k iy s ay d/***	✓	×	✓
	/k ih s ay d/	✓	×	✓
	/k w ay s ay d/	×	✓	✓
	/k w oy s iy d/	×	✓	✓
thyme	***/t ay m/***	✓	×	✓
	/th ay m/	×	✓	✓
	/f ay m/	×	✓	✓
	/th iy m/	×	✓	✓
	/f iy m/	×	✓	✓

Note: ✓ denotes the pronunciation can be implicitly modeled by the corresponding model, whereas × indicates it is difficult to be modeled. The canonical pronunciations are highlighted in bold type.

TABLE II
DETAILS OF CORPORA USED IN OUR EXPERIMENTS

	TIMIT		CU-CHLOE	
	Train	Train	Develop	Test
Speakers	630	147	21	42
Unlabeled	—	67h	—	—
Labeled	4h	26h	4h	7.5h

TABLE III
INTER-ANNOTATOR AGREEMENT FOR PHONETIC TRANSCRIPTIONS

	Ann-1	Ann-2	Ann-3
Ann-2	**0.784** (0.825)	– –	– –
Ann-3	**0.784** (0.813)	**0.752** (0.785)	– –
Ann-4	**0.806** (0.834)	**0.776** (0.794)	**0.800** (0.809)

Note: the numbers in bold type and in parentheses indicate the kappa values over the 48-phone set and 39-phone set [71], respectively.

modeled by the AGM. The AGPM, which incorporates the APM and AGM, can implicitly model both P2LP and G2LP conversions.

VIII. EXPERIMENTAL SETUPS

A. Experimental Data

Our experiments are based on the TIMIT [83]–[85] and CU-CHLOE (**C**hinese **U**niversity **Ch**inese **L**earners **o**f **E**nglish) corpora. The CU-CHLOE corpus contains 110 Mandarin speakers (60 males and 50 females) and 100 Cantonese speakers (50 males and 50 females). There are five parts in CU-CHLOE: confusable words, minimal pairs, phonemic sentences, the Aesop's Fable "The North Wind and the Sun" and prompts from TIMIT. Excluding the TIMIT prompts, all the other parts are labeled by trained linguists, which account for about 30% of the whole CHLOE data.

We randomly select data from 147 speakers as the training set, data from another 21 speakers as the development set and data from the remaining 42 speakers as the test set. The details of the TIMIT and CU-CHLOE corpora are shown in Table II. Note that the data of the TIMIT corpus are native English speech and are also used as part of our training data.

To transcribe the L2 English speech of the CU-CHLOE corpus, we first built acoustic models trained on the TIMIT corpus to align the canonical transcriptions with the L2 English speech. We have trained linguists who annotated the speech with actual pronunciations. To save labor, our linguists mainly focused on labeling (modifying) the phone sequences. For the example showed in Fig. 5, our linguists only revised the phone sequence from $/sil\ dh\ ax\ sil\ n\ ao\ r\ th\ sil/$ to $/sil\ d\ ax\ sil\ l\ ao\ f\ sil/$. Thus the phone boundaries are not changed. Thereafter, these annotated phone sequences are re-aligned using the S-AM described in Section V. We perform forced alignment and train the S-AM iteratively until the S-AM's performance improvements level off, which is assessed via running phonetic recognition on the development set of the CU-CHLOE corpus.

B. Reliability of Manual Annotations

To evaluate the quality of manual annotations, a pilot data set is developed before the CU-CHLOE corpus is collected. This pilot set includes only 108 utterances from 18 Cantonese speakers, none of which is present in the CU-CHLOE corpus described in Table II. Each speaker uttered the same six prompts from the Aesop's Fable "The North Wind and the Sun". Four annotators transcribed this pilot set independently after they are very familiar with the labeling work in CU-CHLOE corpus.

To measure the inter-annotator agreement, the Cohen's Kappa [86]–[89] is employed in this work, given by the following equation:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{15}$$

where p_o is the observed value of agreement among annotators, and p_e is the expected value of agreement by chance. Table III shows that the kappa values for the phonetic transcriptions range from 0.75 to 0.81. A kappa value above 0.75 usually indicates very good reliability [88], [90].

C. DNN Training

The DNN training in this work is similar to those in [7], [62], [69]. In the pre-training stage, all the data (including labeled and unlabeled data) are used to maximize the log-likelihood of RBMs. The one-step Contrastive Divergence [53] is adopted to approximate the stochastic gradient. Ten epochs are performed with a batch size of 512 frames. In the fine-tuning stage, the standard back-propagation algorithm [68] is performed on the labeled data. A dropout [59], [91]–[93] rate of 10% is used in this work. To speed up the BP training process, a technique of asynchronous stochastic gradient descent (ASGD) [94] is used for parallel computing.

TABLE IV
PERFORMANCE OF PHONE RECOGNITION WITH DIFFERENT APPROACHES

	Develop		Test	
	Correct	Acc.	Correct	Acc.
ERN (GMMs) [16]	—	—	79.00%	—
S-AM	80.98%	74.65%	81.15%	74.37%
ERN (S-AM)	86.62%	82.74%	87.02%	83.17%
APM	90.64%	87.68%	90.86%	87.96%
AGM	91.01%	88.67%	91.14%	88.74%
AGPM	**91.66%**	**88.70%**	**91.87%**	**88.92%**
ERN (AGPM)	91.36%	88.13%	91.47%	88.23%

Note: All the above DNNs have four hidden layers of 512 units each. The starting and ending silences are not counted in our experiments.

IX. EXPERIMENTAL RESULTS

A. *Performance of Phone Recognition*

The experimental results of phone recognition are shown in Table IV. The correctness and accuracy are calculated following [95], as shown in (16a) and (16b):

$$\text{Correct.} = \frac{N - S - D}{N} \tag{16a}$$

$$\text{Acc.} = \frac{N - S - D - I}{N} \tag{16b}$$

where N is the total number of labels; while S, D and I denote for the counts of substitution, deletion and insertion errors, respectively.

The ERN approach in [16] achieved a correctness of about 79.0%, where GMMs were used as its acoustic models and the phonological rules generating ERNs are data-driven. Replacing the GMMs with S-AM, the correctness of the ERN approach is greatly improved to 87.0%.

Our baseline system using the S-AM, which is a kind of free-phone recognition, obtains a correctness of 81.2%. Its performance is slightly better than the ERN approach using GMMs as acoustic models, but much worse than the ERN approach with S-AM.

Our APM approach outperforms the above three methods and achieves significant performance improvements. Its correctness and accuracy are 90.9% and 88.0%, respectively. The difference between the S-AM and the APM is that the APM makes use of canonical phone sequences. From canonical and annotated phone sequences, the APM can implicitly learn the phonological transductions, which complement the acoustic features. This method also outperforms the ERN approach with DNNs (i.e., S-AM) as acoustic models, whose P2LP phonological models are trained explicitly and independently from the acoustic models (see Fig. 2(a)).

The AGM implicitly models the G2LP conversion, which is integrated into acoustic modeling. It obtains a correctness of 91.4% and an accuracy of 88.7%, which are slightly better than those of the APM.

The AGPM integrating the APM and AGM achieves the best performance, whose correctness and accuracy are 91.9% and 88.9% respectively. It shows that phonemes and graphemes complement each other.

TABLE V
CONFUSION MATRIX OF MOST FREQUENTLY MISRECOGNIZED VOWELS

		Annotation					
		aa	ah	ae	eh	ih	iy
S-AM	aa	4612	584	91	2	5	0
	ah	299	4280	208	104	476	33
	ae	68	258	1776	476	58	4
	eh	10	84	288	817	84	10
	ih	0	129	25	60	2637	382
	iy	0	29	7	19	613	1772
AGPM	aa	4929	262	64	5	1	0
	ah	176	5113	73	21	194	54
	ae	65	160	2304	98	3	0
	eh	6	32	81	1324	31	5
	ih	1	51	6	28	3870	166
	iy	0	18	0	13	196	2160

TABLE VI
CONFUSION MATRIX OF MOST FREQUENTLY MISRECOGNIZED CONSONANTS

		Annotation					
		d	dh	t	sh	s	z
S-AM	d	1790	308	199	0	1	14
	dh	151	1142	59	1	11	68
	t	317	138	7082	2	31	38
	sh	0	1	12	814	89	2
	s	4	75	71	84	4336	542
	z	2	66	31	4	195	622
AGPM	d	2332	28	86	0	2	4
	dh	204	1917	4	0	2	16
	t	93	9	7725	0	17	7
	sh	0	0	1	850	34	0
	s	0	7	35	59	4622	65
	z	1	2	5	2	113	1303

Applying ERNs to the AGPM, i.e., explicitly constraining the search space in Viterbi decoding, can not gain further improvement. This is due to the disadvantages of the ERN approach, as discussed in Section II-B.

Tables V and VI show the confusion matrices of phones most frequently misrecognized by the S-AM and AGPM. They illustrate that the following confusable pairs are greatly improved by the AGPM: (ae, eh), (ih, iy), (d, dh), (s, z).

B. *Performance of MDD*

As introduced in Section IV , when the recognized phones differ from the canonical transcriptions, mispronunciation detection and diagnosis are achieved respectively. Thus the phone accuracy is one of the most important metrics to evaluate the performance of MDD. In this subsection, we also provide more detailed experimental results of MDD for comparison.

1) Hierarchical Evaluation Structure for MDD: In order to evaluate the performance of MDD, we follow the hierarchical evaluation structure developed in [15] (see Fig. 8), which has also been adopted in [43]. For mispronunciation detection, the

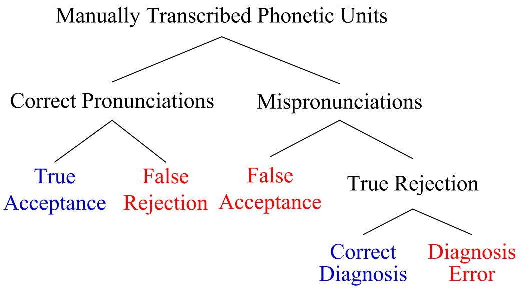


Fig. 8. Hierarchical evaluation structure for mispronunciation detection and diagnosis.

TABLE VII
PERFORMANCE OF MISPRONUNCIATION DETECTION AND DIAGNOSIS WITH DIFFERENT APPROACHES

	FRR	FAR	DER
ERN (GMMs) [16]	15.03%	42.97%	62.00%
ERN (GMMs) [44]	25.63%	22.80%	45.45%
S-AM	22.44%	**15.70%**	23.33%
	(16,651)	(2,282)	(2,858)
ERN (S-AM)	11.04%	43.59%	32.26%
	(8,230)	(5,747)	(2,399)
APM	4.75%	36.61%	15.26%
	(3,478)	(5,592)	(1,477)
AGM	5.25%	31.31%	14.65%
	(3,838)	(4,866)	(1,564)
AGPM	4.57%	30.53%	**13.49%**
	(3,345)	(4,668)	(1,433)
ERN (AGPM)	**1.72%**	62.81%	20.92%
	(1,262)	(8,484)	(1,051)

Note: All the above DNNs have four hidden layers of 512 units each.

correct outcomes are true acceptance (TA) and true rejection (TR), whereas the incorrect outcomes are false rejection (FR) and false acceptance (FA). For mispronunciation diagnosis, we focus on the cases of TR and consider those with diagnostic errors (DE). We can then have the false rejection rate (FRR), false acceptance rate (FAR), and diagnostic error rate (DER) as shown in the following equations:

$$\mathrm{FRR} = \frac{\mathrm{FR}}{\mathrm{TA} + \mathrm{FR}} \tag{17a}$$

$$\mathrm{FAR} = \frac{\mathrm{FA}}{\mathrm{FA} + \mathrm{TR}} \tag{17b}$$

$$\mathrm{DER} = \frac{\mathrm{DE}}{\mathrm{CD} + \mathrm{DE}} \tag{17c}$$

where TA is the number of phones annotated and recognized as correct pronunciation, TR is the number of phones annotated as mispronunciation and identified as incorrect by CAPT systems, FR is the number of phones recognized as mispronunciations when the actual pronunciations are correct, FA is the number of phones misclassified as correct but are actually mispronounced, CD is the number of phones correctly classified as mispronunciations and correctly identified as those that are the same as the annotated phones, and DE is the number of phones correctly identified as mispronunciations but incorrectly diagnosed as those different from the annotated phones.

Table VII shows the results of MDD with different approaches. Note that the ERN approach in [16] worked with data-driven P2LP phonological rules, while the one in [44] made use of canonical transcriptions extracted from dictionaries as well as possible error patterns generated by the G2LP JSM.

The FRR, FAR and DER for the ERN approach in [44] are 25.6%, 22.8% and 45.5%, respectively. The S-AM for free-phone recognition works slightly better, and its FRR, FAR and DER are 22.4%, 15.7% and 23.3%, respectively. Both the ERN and S-AM approaches have large FRRs due to many insertion and substitution errors. Combining the S-AM and the ERNs generated by P2LP phonological models, the FRR is greatly reduced to 11.0%, but the FAR and DER increase. These results indicate that constraining the search space greatly reduces the number of false rejection, and meanwhile the number of true rejection and correct diagnosis are also greatly reduced.

The APM obtains an FRR of about 4.8%, which is much better than the above approaches. The AGM has a similar performance, while the AGPM outperforms the APM and AGM slightly. These are consistent with the results of phone recognition.

Note that the APM has a larger FAR than the S-AM (36.6% vs. 15.7%). One of the reasons is that the APM makes use of acoustic features as well as canonical transcriptions as its input features, and about 84% of the phonemes are correctly pronounced (about 73,000 correct pronunciations vs. about 15,000 mispronunciations, see Table VII). The APM is inclined to recognize a mispronunciation as the correct one (i.e., the input canonical phoneme), provided that the acoustic features are not clearly related to mispronunciation. Similarly, the AGM and AGPM also have large FARs.

Both Tables IV and VII indicate that no further improvement can be achieved by applying ERNs to the AGPM. Although it has a very small value of FRR, its FAR and DER are large.

The metrics of FRR, FAR and DER are widely used in mispronunciation detection and diagnosis, especially the FRR and FAR in mispronunciation detection. However, they still have some problems for evaluation due to the trade-off among them. It is generally accepted that the FRR should be kept as low as possible, since it is usually more unacceptable to identify the correct pronunciations as wrong than to regard the wrong ones as right [96], [97].

2) Other Metrics for MDD: Besides using FRR, FAR and DER to evaluate the performance of MDD, other metrics such as precision, recall and F-measure are also widely used as the performance measures for mispronunciation detection. These metrics are defined as follows:

$$\mathrm{Precision} = \frac{\mathrm{TR}}{\mathrm{TR} + \mathrm{FR}} \tag{18a}$$

$$\mathrm{Recall} = \frac{\mathrm{TR}}{\mathrm{TR} + \mathrm{FA}} = 1 - \mathrm{FAR} \tag{18b}$$

$$\text{F-measure} = 2\,\frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{18c}$$

where TR, FR, and FA are the same as those in (17a–17c). In addition, the accuracies of mispronunciation detection and

TABLE VIII
PERFORMANCE OF MISPRONUNCIATION DETECTION AND DIAGNOSIS

	Mispronunciation Detection				Diagnosis
	Accuracy	Precision	Recall	F-measure	Accuracy
SVM [35]	—	70.0%	44.3%	54.3%	—
SVM [35]	—	80.0%	29.5%	43.1%	—
LR [65]	—	69.2%	69.2%	69.2%	—
S-AM	78.66%	42.39%	**84.30%**	56.41%	76.67%
ERN (S-AM)	84.07%	47.47%	56.41%	51.55%	67.74%
APM	89.75%	73.57%	63.39%	68.10%	84.74%
AGM	90.18%	73.55%	68.69%	71.04%	85.35%
AGPM	**90.94%**	**76.05%**	69.47%	**72.61%**	**86.51%**

Note: All the above DNNs have four hidden layers of 512 units each. In [65], precision is set as the same as recall. The data used in [35], [65] are different from ours.

mispronunciation diagnosis are calculated as follows:

$$\text{For detection:}\quad \text{Accuracy} = \frac{TA + TR}{TA + FR + FA + TR} \qquad (19a)$$

$$\text{For diagnosis:}\quad \text{Accuracy} = \frac{CD}{CD + DE} = 1 - DER \qquad (19b)$$

Table VIII shows the results of MDD measured with the above metrics. For mispronunciation detection, the AGPM can correctly classify the pronunciation as correct or incorrect with an accuracy of 90.9%. Its precision, recall and F-measure are 76.1%, 69.5% and 72.6%, respectively. For mispronunciation diagnosis, its accuracy is 86.5%.

The results of mispronunciation detection in [35], [65] are also given in Table VIII. Although these experimental data are different from ours, it is still meaningful to list their results here for comparison. In [35], a Support Vector Machine (SVM) is used as the classifier for the mispronunciation detection of 13 most frequently mispronounced Mandarin phones. In [65], a neural network based Logistic Regression (LR) classifier is developed to detect the mispronunciation of phones in isolated English words uttered by Chinese learners. In contrast with these tasks that only focus on some most frequently mispronounced phones and phones in isolated words, our work tries to detect all kinds of mispronounced phones in continuous speech and thus should be more difficult.

Moreover, our methods can diagnose the detected mispronunciation, whereas those in [35], [65] can only provide mispronunciation detection.

C. Performance of AGPM With Different Structures

Fig. 9 shows the performance of AGPM with different number of nodes per hidden layer. It shows that the AGPM with only 128 units per hidden layer has already outperformed the S-AM with 512 nodes per hidden layer (see Table IV).

If we increase the units of each hidden layer in AGPM further from 128 to 256, the accuracy is improved greatly from 76.1% to 86.8%. Meanwhile, the real-time factor decreases from 0.52 to 0.50, as more recognition paths with too small probabilities are trimmed in Viterbi decoding. When the number of nodes per hidden layer is larger than 512, the accuracy tends to converge and the AGPM requires much more computing time.

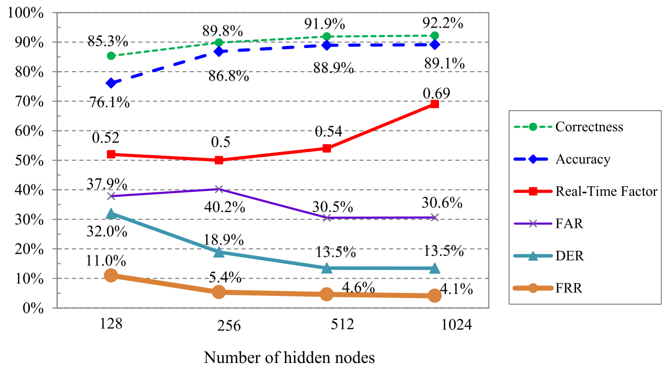


Fig. 9. Performance of AGPM with different number of nodes per hidden layer. All the DNNs have four hidden layers.

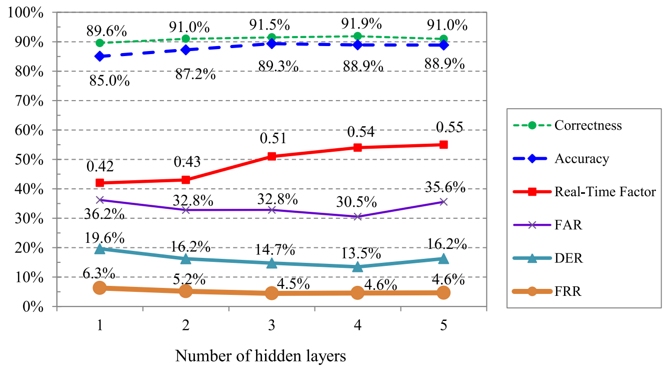


Fig. 10. Performance of AGPM with different number of hidden layers. Each hidden layer has 512 nodes.

Fig. 10 demonstrates how the performance improves as more hidden layers are added to the AGPM. The accuracy is improved from 85.0% with a single hidden layer to 88.9% with four hidden layers.

All the above evaluations were carried out on a Dell Precision T7610 workstation, which has dual Intel Xeon Processor E5-2650 v2 (eight cores of 2.60 GHz). The *jblas* library [98] was used to speed up matrix operations from our Java codes.

X. ANALYSIS AND DISCUSSION

This section presents comparisons among different approaches based on free-phone recognition, ERNs, APM, AGM and AGPM. In addition, we discuss the agreement between the AGPM and each annotator.

To be fair, we only adopt the results of the DNNs with the same hidden structures (i.e., four hidden layers of 512 units each) for comparisons, as given in Tables IV and VII. We prefer this kind of simple hidden structure to illustrate the performance of different approaches, since Section IX-C shows that it has already attained quite good performance for AGPM. With more complex hidden structures, the DNNs require much more training time and can only achieve limited improvements.

A. ERN Approach vs. Free-Phone Recognition

For the approach based on forced alignment using ERNs, its objective function can be presented as (20):

$$\hat{\boldsymbol{s}} = \mathop{\arg\max}_{\mathbf{s} \in \boldsymbol{S}_{\mathbf{ERN}}} p(\mathbf{s} \mid \mathbf{x}) \tag{20}$$

where $\boldsymbol{S}_{\mathbf{ERN}}$ denotes the possible phone-state sequences constrained by ERNs, which can be generated by P2LP phonological rules [16], [20] or G2LP JSM [15].

To identify the phone-state sequence with the highest posterior probability, free-phone recognition explores all possible sequences, as given in (6); whereas the ERN approach only searches the phone-state sequences within $\boldsymbol{S}_{\mathbf{ERN}}$. As the search space is greatly reduced, this kind of approach would outperform free-phone recognition, provided that they use the same acoustic models (see Section IX-A).

B. APM Approach vs. ERN Approach

As discussed in Section II-B, there are still some problems for the ERN approach; whereas our APM method has the following advantages: (1) there is no need to build ERNs, thus it is much easier to be implemented; (2) it incorporates phonological transductions in acoustic modeling using an MD-DNN, hence contextual information is exploited more effectively; (3) it does not explicitly constrain the search space in Viterbi decoding and works with the flexibility like free-phone recognition, thereby any phones can be correctly recognized if the APM is well trained.

C. APM Approach vs. Free-Phone Recognition

Comparing the target functions for free-phone recognition and the APM approach, i.e., (6) vs. (11), we can find that the major difference is whether the approach makes use of canonical transcriptions ($\boldsymbol{q}^{\mathbf{Dict}}$). Since more relevant information is exploited, the APM method attains much better performance.

D. AGM vs. APM

The AGM works similarly to the APM, but implicitly models the G2LP conversion. Since it is more effective to generate likely pronunciations from canonical transcriptions than from graphemes, the AGM works slightly worse than the APM (see Sections IX-A and IX-B). However, the approach using the AGM has an advantage over the one using the APM or ERNs, since the AGM does not require any dictionaries to recognize the phones uttered by L2 learners. It is tedious to update the dictionaries whenever we encounter OOV words—this part is important in practical applications if L2 learners are allowed to design their own prompts for practice. When the canonical transcriptions for OOV words are not available, the approach using the APM or ERNs will fail to recognize the L2 English speech, not to mention mispronunciation detection and diagnosis. On the other hand, the AGM can still recognize the phones uttered by L2 learners and provide feedback of MDD for the words with canonical transcriptions.

TABLE IX
AGREEMENT BETWEEN AGPM AND EACH ANNOTATOR FOR PHONETIC TRANSCRIPTIONS

	Ann-1	Ann-2	Ann-3	Ann-4
AGPM	**0.752** (0.787)	**0.735** (0.746)	**0.758** (0.793)	**0.809** (0.827)

Note: the numbers in bold type and in parentheses indicate the kappa values over the 48-phone set and 39-phone set, respectively.

E. Agreement Between AGPM and Each Annotator

Combining the APM and AGM, the AGPM can implicitly model both P2LP and G2LP conversions. Experiments show that it gains further performance improvements. To evaluate its agreement with each annotator, we calculate the Cohen's Kappa in Table IX. These values are close to those of inter-annotator agreement given in Table III. The mean values of these two tables are 0.764 and 0.784 respectively. Note that there is no overlap of speakers among the pilot set, the training set and the test set (see Section VIII).

XI. CONCLUSION AND FUTURE WORK

In this paper, we investigate mispronunciation detection and diagnosis (MDD) using multi-distribution deep neural networks (MD-DNNs). We first build a State-level Acoustic Model (S-AM) using a DNN to align the canonical transcriptions with second-language (L2) English speech. Then we construct an Acoustic-Phonemic Model (APM) using an MD-DNN to incorporate acoustic MFCC features (modeled by linear units with Gaussian noise) and corresponding canonical transcriptions (represented by binary variables). The AGPM can implicitly model both grapheme-to-likely-pronunciation (G2LP) and phoneme-to-likely-pronunciation (P2LP) conversions, which are integrated into acoustic modeling. A 7-gram State Transition Model (STM), which is also a DNN, is built for Viterbi decoding. With the AGPM and STM, we develop a unified framework for mispronunciation detection and diagnosis, which works with the flexibility like free-phone recognition. Comparing with the approach using Extended Recognition Networks (ERNs), which constrains the recognition paths to some highly possible pronunciations, our method is simpler and more effective. Experimental results show that the free-phone recognition using the S-AM for L2 English speech obtains a phone error rate (PER) of 25.6%. The ERN approach with the S-AM achieves a PER of 16.8%. Our proposed AGPM method gains a significant performance improvement overall, whose PER is 11.1%. For mispronunciation detection and diagnosis, the AGPM achieves a false rejection rate of 4.6%, a false acceptance rate of 30.5% and a diagnostic error rate of 13.5%.

At present, we use the 48 phones as in [71] to transcribe the L2 English speech. This is based on the assumption that non-native speech can be annotated in terms of the categorical phone units from native inventory. However, linguists who have attempted such annotations have clearly pointed out the difficulty in this task because the native phonetic inventory/inventories cannot adequately capture the acoustic characteristics of non-native speech. This motivates us to generate an inventory of L2 phone-like units from observed data in future. We also need to derive the relationships between an L2 phone-like unit and the native phone unit(s). Therefore, when the acoustic decoder selects an L2 phone-like unit, it will imply that a mispronunciation has occurred and the error(s) may be diagnosed based on the mappings between the L2 phone-like unit and the native phone(s).

References

[1] H. Meng, C.-Y. Tseng, M. Kondo, A. Harrison, and T. Viscelgia, "Studying L2 suprasegmental features in Asian Englishes: A position paper," in *Proc. Interspeech*, 2009, pp. 1715–1718.

[2] K. Imoto, Y. Tsubota, A. Raux, T. Kawahara, and M. Dantsuji, "Modeling and automatic detection of English sentence stress for computer-assisted English prosody learning system," in *Proc. 7th Int. Conf. Spoken Lang. Process.*, 2002, pp. 749–752.

[3] F. Tamburini, "Prosodic prominence detection in speech," in *Proc. 7th Int. Symp. Signal Process. Appl.*, 2003, pp. 385–388.

[4] J. Tepperman and S. Narayanan, "Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 937–940.

[5] K. Li, S. Zhang, M. Li, W.-K. Lo, and H. Meng, "Prominence model for prosodic features in automatic lexical stress and pitch accent detection," in *Proc. Interspeech 12th Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 2009–2012.

[6] K. Li and H. Meng, "Perceptually-motivated assessment of automatically detected lexical stress in L2 learners' speech," in *Proc. 8th Int. Symp. Chin. Spoken Lang. Process.*, 2012, pp. 179–183.

[7] K. Li and H. Meng, "Lexical stress detection for L2 English speech using deep belief networks," in *Proc. Interspeech*, 2013, pp. 1811–1815.

[8] K. Li and H. Meng, "Automatic lexical stress and pitch accent detection for L2 English speech using multi-distribution deep neural networks," *Speech Commun.*, submitted for publication.

[9] X.-J. Sun, "Pitch accent prediction using ensemble machine learning," in *Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 953–956.

[10] Y. Ren, S.-S. Kim, M. Hasegawa-Johnson, and J. Cole, "Speaker-independent automatic detection of pitch accent," in *Proc. Speech Prosody*, 2004, pp. 521–524.

[11] J. Zhao, W.-Q. Zhang, H. Yuan, M. T. Johnson, J. Liu, and S. Xia, "Exploiting contextual information for prosodic event detection using auto-context," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, pp. 1–14, 2013.

[12] K. Li, S. Zhang, M. Li, W.-K. Lo, and H. Meng, "Detection of intonation in L2 English speech of native Mandarin learners," in *Proc. 7th Int. Symp. Chin. Spoken Lang. Process.*, 2010, pp. 69–74.

[13] J. P. Arias, N. B. Yoma, and H. Vivanco, "Automatic intonation assessment for computer aided language learning," *Speech Commun.*, vol. 52, no. 3, pp. 254–267, 2010.

[14] H. Meng, Y.-Y. Lo, L. Wang, and W. Y. Lau, "Deriving salient learners' mispronunciations from cross-language phonological comparisons," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2007, pp. 437–442.

[15] X.-j. Qian, H. Meng, and F. Soong, "Capturing L2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (CAPT)," in *Proc. 7th Int. Symp. Chin. Spoken Lang. Process.*, 2010, pp. 84–88.

[16] W.-K. Lo, S. Zhang, and H. Meng, "Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 765–768.

[17] G. Kawai and K. Hirose, "A method for measuring the intelligibility and nonnativeness of phone quality in foreign language pronunciation training," in *Int. Conf. Spoken Lang. Process.*, 1998, pp. 782–785.

[18] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, no. 2, pp. 95–108, 2000.

[19] A. M. Harrison, W.-Y. Lau, H. Meng, and L. Wang, "Improving mispronunciation detection and diagnosis of learners' speech with context-sensitive phonological rules based on language transfer," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008, pp. 2787–2790.

[20] A. M. Harrison, W.-K. Lo, X.-j. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Proc. SLaTE Conf.*, 2009, pp. 45–48.

[21] Y.-B. Wang and L.-S. Lee, "Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 5049–5052.

[22] B. Mak *et al.*, "PLASER: Pronunciation learning via automatic speech recognition," in *Proc. HLT-NAACL Workshop Building Educ. Appl. Using Natural Lang. Process.*, 2003, pp. 23–29.

[23] B. Dong, Q. Zhao, J. Zhang, and Y. Yan, "Automatic assessment of pronunciation quality," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, 2004, pp. 137–140.

[24] W. Liang, J. Liu, and R. Liu, "Automatic spoken English test for Chinese learners," in *Proc. Int. Conf. Commun., Circuits Syst.*, 2005, p. 860.

[25] C.-L. Li, J. Liu, and S.-H. Xia, "Perceptual evaluation of pronunciation quality for computer assisted language learning," in *Proc. 1st Int. Conf. Technol. E-Learn. Digit. Entertainment*, 2006, pp. 17–26.

[26] Q. Shi, K. Li, S. Zhang, S. M. Chu, J. Xiao, and Z. Ou, "Spoken English assessment system for non-native speakers using acoustic and prosodic features," in *Proc. Interspeech*, 2010, pp. 1874–1877.

[27] A. Lee and J. Glass, "A comparison-based approach to mispronunciation detection," in *Proc. SLaTE Conf.*, 2012, pp. 382–387.

[28] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8227–8231.

[29] A. Lee and J. Glass, "Context-dependent pronunciation error pattern discovery with limited annotations," in *Proc. Interspeech*, 2014, pp. 2877–2880.

[30] L. Neumeyer, H. Franco, M. Weintraub, and P. Price, "Automatic text-independent pronunciation scoring of foreign language student speech," in *Proc. 4th Int. Conf. Spoken Lang.*, 1996, pp. 1457–1460.

[31] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1997, pp. 1471–1474.

[32] H. Franco *et al.*, "Eduspeak: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications," *Lang. Test.*, vol. 27, no. 3, pp. 401–418, 2010.

[33] J.-C. Chen, J.-S. Jang, J.-Y. Li, and M.-C. Wu, "Automatic pronunciation assessment for Mandarin Chinese," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2004, pp. 1979–1982.

[34] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic scoring of pronunciation quality," *Speech Commun.*, vol. 30, no. 2, pp. 83–93, 2000.

[35] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," *Speech Commun.*, vol. 51, no. 10, pp. 896–905, 2009.

[36] C. Cucchiarini, F. De Wet, H. Strik, and L. Boves, "Assessment of Dutch pronunciation by means of automatic speech recognition technology," in *Proc. 5th Int. Conf. Spoken Lang. Process.*, 1998, pp. 751–754.

[37] M. Nicolao, A. V. Beeston, and T. Hain, "Automatic assessment of English learner pronunciation using discriminative classifiers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5351–5355.

[38] J. van Doremalen, C. Cucchiarini, and H. Strik, "Using non-native error patterns to improve pronunciation verification," in *Proc. Interspeech*, 2010, pp. 590–593.

[39] A. Lee and J. Glass, "Pronunciation assessment via a comparison-based system," in *Proc. IEEE SLaTE Workshop*, 2013, pp. 122–126.

[40] Y. Kim, H. Franco, and L. Neumeyer, "Automatic pronunciation scoring of specific phone segments for language instruction," in *Proc. 5th Eur. Conf. Speech Commun. Technol.*, 1997, pp. 645–648.

[41] O. Ronen, L. Neumeyer, and H. Franco, "Automatic detection of mispronunciation for language instruction," in *Proc. Eur. Conf. Speech Commun. Technol.*, 1997, pp. 649–652.

[42] Y.-B. Wang and L.-S. Lee, "Toward unsupervised discovery of pronunciation error patterns using universal phoneme posteriorgram for computer-assisted language learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8232–8236.

[43] Y. Wang and L. Lee, "Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 3, pp. 564–579, Mar. 2015.

[44] X.-J. Qian, H. Meng, and F. Soong, "On mispronunciation lexicon generation using joint-sequence multigrams in computer-aided pronunciation training (CAPT)," in *Proc. Interspeech*, 2011, pp. 865–868.

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

[46] K. Truong, A. Neri, C. Cucchiarini, and H. Strik, “Automatic pronunciation error detection: An acoustic-phonetic approach,” in *Proc. InSTIL/ICALL*, 2004, pp. 3040–3051.

[47] J. Tepperman and S. Narayanan, “Using articulatory representations to detect segmental errors in nonnative pronunciation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 8–22, Jan. 2008.

[48] P. Ladefoged and K. Johnson, *A Course in Phonetics*. Boston, MA, USA: Cengage Learning, 2014.

[49] M. Richardson, J. Bilmes, and C. Diorio, “Hidden-articulator Markov models for speech recognition,” *Speech Commun.*, vol. 41, no. 2, pp. 511–529, 2003.

[50] T. J. Hazen, W. Shen, and C. White, “Query-by-example spoken term detection using phonetic posteriorgram templates,” in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2009, pp. 421–426.

[51] Y. Zhang and J. R. Glass, “Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams,” in *IEEE Workshop Autom. Speech Recognit. Understanding*, 2009, pp. 398–403.

[52] M. A. Carlin and M. Elhilali, “Exploiting temporal coherence in speech for data-driven feature extraction,” in *Proc. 45th Annu. Conf. Inf. Sci. Syst.*, 2011, pp. 1–5.

[53] G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, pp. 1527–1554, 2006.

[54] G. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, pp. 504–507, 2006.

[55] A.-r. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, “Deep belief networks using discriminative features for phone recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 5060–5063.

[56] A.-r. Mohamed, G. E. Dahl, and G. E. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[57] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large vocabulary speech recognitio,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[58] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4277–4280.

[59] L. Deng, O. Abdel-Hamid, and D. Yu, “A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6669–6673.

[60] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.

[61] L. Deng and J. C. Platt, “Ensemble deep learning for speech recognition,” in *Proc. Interspeech*, 2014, pp. 1915–1919.

[62] X.-j. Qian, H. Meng, and F. Soong, “The use of DBN-HMMs for mispronunciation detection and diagnosis in L2 English to support computer-aided pronunciation training,” in *Proc. Interspeech*, 2012, pp. 755–758.

[63] W. Hu, Y. Qian, and F. Soong, “A new DNN-based high quality pronunciation evaluation for computer-aided language learning (CALL),” in *Proc. Interspeech*, 2013, pp. 1886–1890.

[64] W. Hu, Y. Qian, and F. Soong, “A new neural network based logistic regression classifier for improving mispronunciation detection of L2 language learners,” in *Proc. 9th Int. Symp. Chin. Spoken Lang. Process.*, 2014, pp. 245–249.

[65] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, “Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers,” *Speech Commun.*, vol. 67, pp. 154–166, 2015.

[66] S. Kang, X. Qian, and H. Meng, “Multi-distribution deep belief network for speech synthesis,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8012–8016.

[67] S. Kang and H. Meng, “Statistical parametric speech synthesis using weighted multi-distribution deep belief network,” in *Proc. Interspeech*, 2014, pp. 1959–1963.

[68] D. E. Rumelhart, G. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[69] K. Li and H. Meng, “Mispronunciation detection and diagnosis in L2 English speech using multi-distribution deep neural networks,” in *Proc. 9th Int. Symp. Chin. Spoken Lang. Process.*, 2014, pp. 255–259.

[70] K. Li, X. Qian, S. Kang, P. Liu, and H. Meng, “Integrating acoustic and state-transition models for free phone recognition in L2 English speech using multi-distribution deep neural networks,” in *Proc. SLaTE Conf.*, 2015, pp. 119–124.

[71] K.-F. Lee and H.-W. Hon, “Speaker-independent phone recognition using hidden Markov models,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 11, pp. 1641–1648, Nov. 1989.

[72] R. M. Kaplan and M. Kay, “Regular models of phonological rule systems,” *Comput. Linguistics*, vol. 20, no. 3, pp. 331–378, 1994.

[73] M. J. Dedina and H. C. Nusbaum, “Pronounce: A program for pronunciation by analogy,” *Comput. Speech Lang.*, vol. 5, no. 1, pp. 55–64, 1991.

[74] W. M. Daelemans and A. P. Van den Bosch, “Language-independent data-oriented grapheme-to-phoneme conversion,” in *Progress in Speech Synthesis.* New York, NY, USA: Springer, 1997, pp. 77–89.

[75] S. F. Chen, “Conditional and joint models for grapheme-to-phoneme conversion,” in *Proc. Interspeech*, 2003, pp. 2033–2036.

[76] J. R. Bellegarda, “Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy,” *Speech Commun.*, vol. 46, no. 2, pp. 140–152, 2005.

[77] P. Taylor, “Hidden Markov models for grapheme to phoneme conversion,” in *Proc. 9th Eur. Conf. Speech Commun. Technol.*, 2005, pp. 1973–1976.

[78] H. Meng, S. Hunnicutt, S. Seneff, and V. Zue, “Reversible letter-to-sound/sound-to-letter generation based on parsing word morpology,” *Speech Commun.*, vol. 18, no. 1, pp. 47–63, 1996.

[79] T. J. Sejnowski and C. R. Rosenberg, “Parallel networks that learn to pronounce English text,” *Complex Syst.*, vol. 1, no. 1, pp. 145–168, 1987.

[80] X. Gonzalvo and M. Podsiadło, “Text-to-speech with cross-lingual neural network-based grapheme-to-phoneme models,” in *Proc. Interspeech*, 2014, pp. 765–769.

[81] J. A. Bullinaria, “Text to phoneme alignment and mapping for speech technology: A neural networks approach,” in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 625–632.

[82] E. B. Bilcu, “Text-to-phoneme mapping using neural networks,” Ph.D. dissertation, Tampere Univ. Technol., Tampere, Finland, 2008. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-200902201017

[83] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, “The Darpa speech recognition research database: Specifications and status,” in *Proc. DARPA Workshop Speech Recognit.*, 1986, pp. 93–99.

[84] L. F. Lamel, R. H. Kassel, and S. Seneff, “Speech database development: Design and analysis of the acoustic-phonetic corpus,” in *Proc. ESCA Tut. Res. Workshop Speech Input/Output Assessment Speech Databases*, 1989, pp. 161–170.

[85] V. Zue, S. Seneff, and J. Glass, “Speech database development at MIT: TIMIT and beyond,” *Speech Commun.*, vol. 9, no. 4, pp. 351–356, 1990.

[86] J. Cohen, “A coefficient of agreement for nominal scales,” *Educ. Psychosocial Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

[87] J. Carletta, “Assessing agreement on classification tasks: The Kappa statistic,” *Comput. Linguistic*, vol. 22, no. 2, pp. 249–254, 1996.

[88] L. Dilley, M. Breen, M. Bolivar, J. Kraemer, and E. Gibson, “A comparison of inter-transcriber reliability for two systems of prosodic annotation: RaP (Rhythm and Pitch) and ToBI (Tones and Break Indices),” in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 317–320.

[89] M. Breen, L. C. Dilley, J. Kraemer, and E. Gibson, “Inter-transcriber reliability for two systems of prosodic annotation: ToBI (Tones and Break Indices) and RaP (Rhythm and Pitch),” *Corpus Linguist. Linguistic Theory*, vol. 8, no. 2, pp. 277–312, 2012.

[90] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical Methods for Rates and Proportions*. New York, NY, USA: Wiley, 2013.

[91] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[92] M. L. Seltzer, D. Yu, and Y. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7398–7402.

[93] A. Hannun *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567v2*, 2014.

[94] S. Zhang, C. Zhang, Z. You, R. Zheng, and B. Xu, “Asynchronous stochastic gradient descent for DNN training,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6660–6663.

[95] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, “The HTK book (for HTK version 3.4),” Cambridge Univ. Eng. Dept., 2006, pp. 279–281.

[96] L. F. Bachman and S. J. Savignon, *Fundamental Considerations in Language Testing.*, Oxford, U.K.: Oxford Univ. Press, 1990.

[97] M. Eskenazi, “An overview of spoken language technology for education,” *Speech Commun.*, vol. 51, no. 10, pp. 832–844, 2009.

[98] M. L. Braun, *Linear algebra for java (version 1.2.3)*, 2010. [Online]. Available: http://jblas.org/

Kun Li received the B.E. degree from Zhejiang University, Hangzhou, China, in 2006, and the M.E. degree from Tsinghua University, Beijing, China, in 2009. He joined the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, as a Research Assistant in 2009 and then as a Ph.D. candidate in 2011. His research interests include segmental and suprasegmental mispronunciation detection and diagnosis in nonnative English speech.

Xiaojun Qian received the B.E. degree in electrical engineering from Fudan University, Shanghai, China, in 2007. From 2007 to 2010, he was with the speech group of Microsoft Research Asia. He joined the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, as a Ph.D. candidate in 2009. His research interests include discriminative training, subspace acoustic modeling, and deep learning. He received the 2010 Microsoft Research Asia Fellowship Award.

Helen Meng (F'13) received the S.B., S.M., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. She joined The Chinese University of Hong Kong in 1998, where she is currently a Professor and Chair with the Department of Systems Engineering and Engineering Management. She was also the Associate Dean of Research of the Faculty of Engineering between 2005 and 2010. Her research interests include human–computer interaction via multimodal and multilingual spoken language systems, speech retrieval technologies, and computer-aided pronunciation training. She served as the Editor-in-Chief of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING from 2009 to 2011. She was an Elected Board Member of the International Speech Communication Association as well as an Elected Member of the IEEE Signal Processing Society Board of Governors. She is a Fellow of the HKCS, HKIE, and ISCA.