INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Recognize Mispronunciations to Improve Non-native Acoustic Modeling through a Phone Decoder Built from One Edit Distance Finite State Automaton

Wei Chu[1], *Yang Liu*[2], *Jianwei Zhou*[3]

[1]PAII Inc., USA
[2]Amazon, USA
[3]LAIX Inc.

chuwei129@pingan.com.cn, yangliud@amazon.com, andrea.zhou@liulishuo.com

Abstract

This paper proposed a procedure for detecting and recognizing mispronunciations in training data, and improved non-native acoustic modeling by training with the corrected phone alignments. To start, an initial phone sequence for an utterance is derived from its word-level transcription and a dictionary of canonical pronunciation. Following that, the region of mispronunciation is detected through examining phone-level goodness-of-pronunciation (GOP) scores. Then over the region, a constrained phone decoder is used to recognize the most likely pronounced phone sequence from all the possible phone sequences with one phone edit distance from the initial phone sequence. After updating the phone alignments and GOP scores, this detection and recognition procedure is repeated until no more mispronunciation is detected. Experiments on a 300-hour non-native spontaneous dataset showed that the acoustic model trained from the proposed procedure reduced WER by 6% compared to a well optimized context-dependent factorized-TDNN HMM baseline system with the same neural network topology. This work also offered a data-driven approach for generating a list of common mispronunciation patterns of non-native English learners that may be useful for speech assessment purpose.

Index Terms: acoustic modeling, goodness-of-pronunciation, mispronunciation detection and recognition, phone decoding

1. Introduction

Prior work has been conducted for automatic assessment using state-of-the-art acoustic modeling algorithms in the domain of English as a Second Language (ESL) learning [1, 2, 3]. Besides assessing the goodness of pronunciation, it is also useful to automatically transcribe the accented speech into text, which serves as an input to various subsequent applications, such as grammar correction and teacher-student dialogue systems. It has been shown in the past work that recognizing accented speech, especially the speech from an ESL learner is not an easy task [4, 5, 6, 7, 8].

ESL learners tend to make mispronunciation beside accented pronunciations in the space of phones defined by a canonical dictionary [9, 10]. Early work on mispronunciation detection includes [11]. A decoding network, i.e., Extended Recognition Network (ERN) [12], designed by experienced linguists can be used to recognize mispronunciations including phone substitutions, deletions, and insertions. Phonological rules were automatically derived from human phonetic transcriptions for detecting mispronunciations, however, the detection error increased significantly when using transcriptions generated by ASR [13]. Dynamic time warping was used to find the mis-alignments that are then used as features in the following Support Vector Machine (SVM) based mispronunciation detector [14]. Sub-segmental speech attributes like manner and place of articulation were used for better detecting mispronunciations [15]. Works on mispronunciation detection have been extended to the diagnosis including the recognition of the mispronunciations using SVM [16] or deep learning [17].

In this work, our main goal is to improve the recognition accuracy of non-native speech. It is natural to consider employing an ERN-based alignment approach to detect and recognize the mispronounced phones, and then generate GMM alignments that are not aligned to wrong phone states as before, such that a subsequent neural network training could benefit from the corrected alignments.

Since an ERN can not be designed when linguists are absent, and using an unconstrained phone decoder without strong linguistic guidance has been proven to be error-prone even on native read speech [18] [19], in this work, a data-driven approach is proposed. We first detect the mispronunciations by leveraging the Goodness-of-Pronunciation (GOP) score [20] from speech assessment, then use a specially designed constrained phone decoder with a low complexity in the search space to recognize the mispronunciations and correct the original phone alignments. This way the subsequent neural network based acoustic model (AM) training can be improved since acoustic features of mispronounced phone sequences can now be associated with the correct phone labels.

2. The Proposed AM Training

2.1. System Overview

A system diagram for training the proposed non-native acoustics model is shown in Fig. 1. A standard baseline system uses only the non-native speech to train the AM, i.e., GMM pre-training to obtain alignments, which are then used for subsequent Time Delayed Neural Networks (TDNN) training [21]. Our method leverages native speech to train a native GMM to obtain GOP scores that are further used to generate better alignments for non-native AM training. The components shown in the diagram are all standard ones except the two rounds of the 'GOP-based GMM alignment' procedure, which will be explained in details in the following sections.

This work was done while the first two authors were at LAIX Inc.

http://dx.doi.org/10.21437/Interspeech.2020-3109

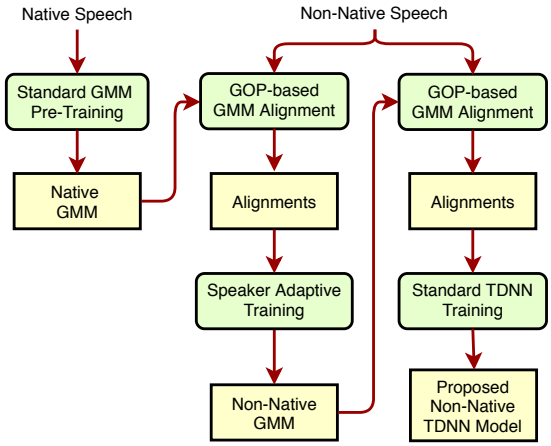

Figure 1: *A system flowchart of the proposed method for training a non-native acoustic model.* ***GOP****: Goodness-of-Pronunciation,* ***TDNN****: Time-Delayed Neural Networks.*

2.2. The Proposed AM Training Procedure

Our proposed method shown in Fig. 1 has the following components :

• **Step 1**: *Training Native GMM*

Employ the standard GMM pre-training process to obtain a GMM and a decision tree on native speech.

• **Step 2**: *GOP-based Mispronunciation Detection with Native GMM*

For a non-native spoken utterance and its generated phone sequence according to the word-level transcription and canonical dictionary, run GMM alignment with the native GMM, and then find the phone y_i with the lowest GOP score GOP(y_i). If GOP$(y_i) < 0$, proceed to Step 3; otherwise proceed to the next utterance. If all the utterances have been processed, move to Step 4.

The GOP score [20] in a GMM-HMM based framework is calculated as $\text{GOP}(y) \approx \frac{1}{N_p} \log \frac{p(\mathbf{x}|y)P(y)}{\max_z p(\mathbf{x}|z)P(z)}$, where y denotes the phone to be read, x denotes the segment of features to which phone y is aligned, N_p denotes the duration of $\mathbf{x}$ in frames, z denotes another competing phone. Note that GOP$(y) \leq 0$.

• **Step 3**: *Constraint Mispronunciation Recognition with Native GMM*

Based on phone y_i, a minimized FSA of phones denoted by $L_{partial}$ is created, as shown in Fig. 2. Note that we use word-position-dependent phones. Then a decoding graph $HCL_{partial}$ is composed and minimized, where H and C denote the HMM-state and phonetic context transducers trained from native speech, i.e., the native GMM and decision tree.

This FSA can accept substitution errors (path $0 \rightarrow 1 \rightarrow 5 \rightarrow 3$ as in Fig. 2), deletion errors ($0 \rightarrow 1 \rightarrow 3$), and left insertion ($0 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 3$) and right insertion ($0 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3$) errors. Note this FSA does not have a loop-back arc from the final state to the initial state as the common FSA used in a phone decoder.

Then using this new phone FSA, we obtain the 1-best decoding result, i.e., a new phone sequence denoted by $\mathbf{y}_n$, for the features aligned to the original phone sequence $\mathbf{y}_o$, i.e., y_{i-1}, y_i, y_{i+1} in Step 2. GOP scores are then calculated for the phones in the new phone sequence. A sequence GOP (S-GOP) score is defined as a criterion for when to stop searching

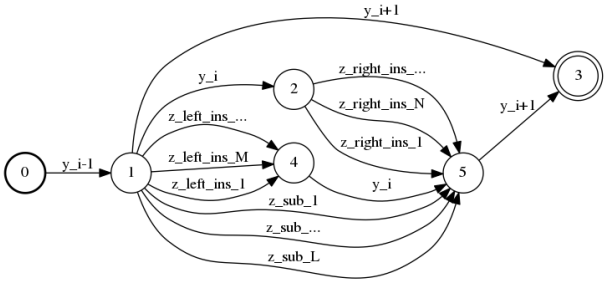

Figure 2: *An FSA of word-position-dependent phones for searching possible pronunciation errors of deletion, insertion, and substitution.* y_{i-1} *and* y_{i+1}*, denote left and right neighbor phones of the center phone* y_i*.* L*,* M*, and* N *are the numbers of phone candidates for searching left insertion, right insertion, and substitution, respectively. Deletion error searching is done by skipping* y_i *arc. The omitted weight on each arc is 1.0.*

for mispronunciations:

$$\text{S-GOP}(\mathbf{y}) = \frac{\sum_j N_j \cdot \text{GOP}(y_j)}{\sum_j N_j} \tag{1}$$

where N_j denotes the duration of the j^{th} non-silence phone in $\mathbf{y}$ in frames. If the following condition holds,

$$\frac{\text{S-GOP}(\mathbf{y}_n) - \text{S-GOP}(\mathbf{y}_o)}{|\text{S-GOP}(\mathbf{y}_o)|} > \alpha \tag{2}$$

where α is an empirically set threshold with a positive value, we replace the old phone sequence and its phone GOP scores with the new ones in the alignment results, and repeat Step 3; otherwise return to Step 2 to process the next utterance.

• **Step 4**: *Speaker Adaptive Training of Non-Native GMM*

Train a non-native GMM and decision tree in a speaker adaptive training style using the final alignments obtained from the end of Step 2, then proceed to Step 5.

• **Step 5**: *GOP-based Mispronunciation Detection with Non-Native GMM*

Perform the same procedure as described in Step 2 except that non-native GMM obtained in Step 4 are used here. If GOP$(y_i) < 0$, proceed to Step 6; otherwise proceed to the next utterance. If all the utterances have been processed, jump to Step 7.

• **Step 6**: *Constraint Mispronunciation Recognition with Non-Native GMM*

Perform the same procedure as described in Step 3 but using non-native GMM obtained in Step 4. If the S-GOP score indicates that searching is not over yet, then repeat Step 6; otherwise return to Step 5 to process the next utterance.

• **Step 7**: *Neural Network AM Training with Alignments*

Train a non-native TDNN model using the alignments obtained from the end of Step 5.

3. Experiments

3.1. Data Sets

We use an in-house data set containing native read speech and non-native spontaneous speech. The training set consists of 284 hours spontaneous speech from 2.8k ESL learners whose native language is Chinese (127k utterances in total). An auxiliary dataset of 644 hours read speech from 1.3k native speakers (491k utterances in total) is only used to train a native GMM for

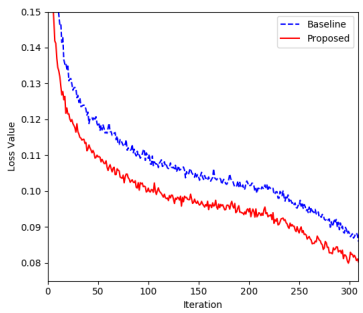

Figure 3: *A comparison of the loss values over iterations for the baseline and our proposed method during TDNN trainings.*

GOP scoring. The test set is composed of 3k utterances from 56 non-native speakers. There are 10 words on average in each utterance.

3.2. ASR Results

3.2.1. Baseline ASR system settings

The following settings are used in our experiments. The CMU dictionary and a G2P model learned from it are used to obtain the pronunciations of a vocabulary of 22K words. 13 dimensional MFCC features are used in GMM training, and 40 Filterbank features are used in TDNN training. No data augmentation is used. The decoder uses an external trigram language model (LM) trained from web text with a pruning threshold of 1e-8. The LM weight is 10 and word insertion penalty is 0, chosen based on our empirical results in similar testing conditions. Our focus is on AM and thus we did not perform higher order n-gram or LSTM/RNN LM rescoring.

We used the Resnet-style TDNN-F model of 11 layers [21, 22] with sequence training (lattice-free MMI) [23]. The output dimension of the factorized and TDNN layers is set to 256 and 1280, respectively. The final output dimension is 3.5K. Note that the numbers of senones at the final GMM training is 5.5K. The initial and final learning rates are set to 1e-3 and 1e-4, respectively. The number of epochs is 6. The L2 regularization factors for non-output layers and the final output layer are set to 0.002 and 0.0005, respectively. No dropout is used.

We spent some efforts on changing the number of layers, the number of neurons in each layer, the learning rates, and regularization factors, and dropout rates, but did not find other better settings .

3.2.2. Settings in our proposed method

The proposed AM used the same GMM and TDNN model topology, and learning parameters as the baseline system. For AM training, we use two rounds of GOP-based alignment, mainly because the speaker adaptive training after the first round generated better LDA and fMLLR transforms, and using non-native GMM in the second round is able to yield alignments with more accurate phone boundaries for the correctly pronounced phone sequence compared to using native GMM.

3.2.3. Results

We first present some training results. Fig. 3 shows the comparison of the loss function changes of our proposed and the baseline systems during TDNN trainings. It can be seen that on the training set, our proposed method has a faster loss drop and a lower final loss compared to the baseline. The results also showed that our proposed method has a lower average loss of the last 5 iterations on the validation set than the baseline (0.081 vs. 0.107). In our proposed method, the difference between the average final loss on the training and validation sets is also smaller compared to the baseline (0.008 vs. 0.020). It is also worth noting that in GMM pre-training, our proposed method also had a significant increase in the final overall likelihood of the GMM over the training data (-41.4 vs. -44.1).

Table 1: *A comparison of WERs(%) and error breakdowns of the baseline and proposed method.* ***SUB/DEL/INS****: substitution/deletion/insertion (with rounding errors).* ***B&P Lattice Fusion****: interpolate lattices from the baseline and proposed method with equal weights.*

	WER	SUB	DEL	INS
Baseline	16.3	9.0	3.7	3.6
Proposed	16.0	8.9	3.7	3.3
B&P Lattice Fusion	15.4	8.5	3.7	3.3

The final WER results of the baseline and the proposed method are shown in Table 1. The proposed method had a 2% drop in WER, mostly because of the 8% drop in insertion errors. The fusion method shown in the last row of the table is based on lattice interpolation with equal weights for the two systems. This reduced the overall WER by 6%, with 6% and 8% drops in substitution and insertion errors, respectively. The improved results from the combination method suggest that the two systems have some different error patterns.

3.3. Analysis of GOP Alignment Results

For all the phone instances that appeared in the training text, the detected mispronunciation rate is 3.8%. Note that the actual mispronunciation rate might be higher, since the threshold α in Eq. 2 was set to a relatively high value of 0.2 in order to focus only on the cases where the recognized mispronunciations result in an higher increase of the GOP scores compared to the original GOP scores. In addition, it may also be because that the training set is selected from the speakers with higher speech assessment scores. The percentage of substitution, insertion, and deletion mispronunciations is 75.4%, 23.6%, and 1.0%, respectively. Since the training data is selected from a large ESL corpus, and only recordings with relatively high assessment scores are selected as the training set used in this work, the lack of deletion mispronunciations might be because of the assessment algorithm we used, which tends to punish deletion errors more in scoring than other mispronunciations.

For analysis, the top-10 detected and recognized mispronunciations with their probabilities in substitution, insertion, and deletion categories are listed in Table 2. It can be seen that it is more common for a vowel to be mispronounced than consonants given the fact that the priors of the vowels and consonants are generally the same in magnitude. For insertion mispronunciations, another vowel can be inserted to the neighbour of the original vowel, which might not be common in the errors that Chinese English learners are prone to make according to linguists [9, 10]. It is also worth noting that those mispronunciation patterns derived from hundreds of hours of speech may reveal the underlying statistics of the Chinese English learners and serve as a promising source of information for providing feedback in speech assessment.

Table 2: *The top-10 mispronunciations of 3 categories: substitution, insertion, and deletion.* ***Transcription****: the phone sequence derived from the word-level transcription and CMU pronunciation dictionary.* ***Mispronunciation****: recognized phone sequence from the features aligned to* ***Transcription*** *using the phone FSA shown in Fig. 2 .* ***P****:* $P(\textit{Transcription}, \textit{Mispronounced}|\textit{Category})$*.* ***K_B****: word-****B****eginning phone* ***K. B/I/E/S****: word-beginning/word-inner/word-ending/word-singleton phone.*

Substitutions			Insertions			Deletions		
Transcription	Mispronunciation	P(%)	Transcription	Mispronunciation	P(%)	Transcription	Mispronunciation	P(%)
K_B AO_I S_E	K_B IH_I S_E	0.16	Y_B AO_I NG_E	Y_B UW_I AO_I NG_E	0.49	UW_I ZH_I AH_I	UW_I AH_I	6.43
D_I IH_I S_I	D_I IY_I S_I	0.14	R_B AH_I B_I	R_B OW_I AH_I B_I	0.27	N_B AY_I IY_I	N_B IY_I	4.24
F_B EH_I NG_E	F_B OW_I NG_E	0.14	G_B IY_E	G_B W_I IY_E	0.20	T_I AH_I R_I	T_I R_I	3.11
W_B AE_I NG_E	W_B AO_I NG_E	0.14	JH_I IH_I T_E	JH_I IH_I NG_I T_E	0.18	EH_I R_I M_I	EH_I M_I	2.76
IY_I EY_I M_I	IY_I AE_I M_I	0.14	T_I AH_I K_E	T_I AA_I AH_I K_E	0.18	JH_I IY_I AA_I	JH_I AA_I	2.30
D_I EH_I K_I	D_I AE_I K_I	0.13	L_B AA_I K_E	L_B AA_I L_I K_E	0.18	N_I L_I AE_I	N_I AE_I	1.72
IH_B SH_I	IY_B SH_I	0.13	IH_B D_I	IH_B IY_I D_I	0.18	AA_I R_I L_I	AA_I L_I	1.62
ER_I DH_I IY_E	ER_I TH_I IY_E	0.12	K_I ER_I IH_I	K_I AA_I ER_I IH_I	0.17	Z_I IY_I AH_I	Z_I AH_I	1.57
L_B IH_I K_I	L_B IY_I K_I	0.12	L_I OY_I ER_E	L_I OY_I Y_I ER_E	0.17	AY_B L_I AH_I	AY_B AH_I	1.38
JH_B AY_I N_I	JH_B IY_I N_I	0.12	L_B EH_I K_I	L_B N_I EH_I K_I	0.17	HH_I W_I EH_I	HH_I EH_I	1.27

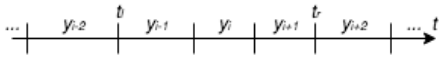

Figure 4: *An illustration of a detected mispronounced phone y_i and its neighbours aligned to features along time. Mispronunciation recognition is performed on a sequence of features between t_l and t_r.*

3.4. Discussions

3.4.1. Reasons for not using an unconstrained phone decoder

We believe it is important to explain why this special style of constraint no-loop-back phone decoder is proposed and used, other than an unconstrained loop-back phone (ULBP) decoder with an optional phone LM.

First, ULBP decoders have been reported to be error-prone. For example, on TIMIT native read speech dataset, the phone error rate of the state-of-the-art systems with very sophisticated architectures is still above 16% [18, 19]. It is questionable whether applying these technologies to non-native spontaneous speech would achieve satisfying results.

Second, the goal of this work is to first detect and then recognize the mispronunciations, not to recognize every phone the speaker said.

Third, an ULBP decoder would still have potential issues of low accuracy even if it runs on the features aligned to the mispronounced phone. For phone deletion errors, the mispronounced phone could still be aligned to a segment of features, making it impossible for an ULBP decoder to recognize the features as no phone; for substitution and insertion errors, because the adjacent phones serve as the context of the mispronounced phone, the alignment may end up with inaccurately estimated boundaries, and thus making it difficult to correctly decode the phone or phone sequence.

3.4.2. Properties of the proposed phone decoder

In this work, we made two assumptions in the phone decoder we used. First, the neighbour phones of a mispronounced phone y_i, i.e., y_{i-2}, y_{i-1}, y_{i+1}, y_{i+2}, are not mispronounced since the mispronunciation rate on this dataset is only 3.8%. Base on that assumption, the boundaries of features used for recognizing mispronunciations, i.e., t_l and t_r as shown in Fig. 4, could be regarded as accurate since the phones on the left and right side of each boundary are not mispronounced. Mispronunciation recognition is over the features between the assumed accurately estimated boundaries, i.e., t_l and t_r, as shown in Fig. 4.

Second, when searching for mispronunciations, only those phone sequences with just one edit distance from the target one are considered. Therefore, we proposed the constrained phone FSA without loop-back arc shown in Fig. 2. The complexity of the search space of the phone sequence is $O(N)$ where N is the size of the phone set. For an ULBP decoder, the complexity is $O(N^M)$ where M is the average number of phones recognized. Because of the existence of the loop-back arc, it is not easy to directly control the variation of M. This assumption reduces the search space of the phone sequences.

It is worth pointing out some issues related to the proposed phone decoder. First, the original y_i is still employed when searching for the left or right insertion, making it easier to detect the insert errors compared to an ULBP decoder where the information of the original y_i is discarded.

Second, since the proposed procedure is an iterative method, after recognizing the mispronunciation at y_i and update the alignments, it is possible for the algorithm to select y_i's neighbour y_{i-1} or y_{i+1}, which may have the second lowest GOP, to repeat this searching process. This indeed loosens the first assumption that sometimes is too strong.

Third, all the weights on each arc of the proposed FSA are set to 1. No change in the final WER has been observed if $HCL_{partial}$ is further composed with a phone FSA derived from a 3-gram phone LM trained on the training text. The final WER became slightly worse than the baseline when using an ULBP decoder, which could be either because no fine tuning on the decoding was conducted, or because it does not work as well as the proposed phone decoder.

Finally, DNN-based GOP [1], and Forced-GOP algorithms [24] have been reported to yield better GOP scores than GMM-based ones. We used GMM-based alignments, since the GOP score is only used as an intermediate indicator for mispronunciation detection, and GMM-based alignments are still commonly used as input for TDNN training.

4. Conclusions

This work proposed a procedure to detect mispronunciations using a GOP scorer, and recognize mispronunciations on the training data using a constrained no-loop-back phone decoder that is low in search space complexity. With the help of the corrected phone alignment on a non-native training set, the final system reduced the WER by 2% compared to a context-dependent factorized-TDNN HMM AM with the same neural network model topology, and reduced the WER by 6% when combining with the baseline using a simple lattice interpolation strategy. This work also offered a data-driven approach for generating a list of common substitution, insertion, and deletion mispronunciation patterns by ESL learners that may be useful for future speech assessment purpose.

5. References

[1] W. Hu, Y. Qian, and F. K. Soong, "A new dnn-based high quality pronunciation evaluation for computer-aided language learning (call)." in *Interspeech*, 2013, pp. 1886–1890.

[2] J. Cheng, X. Chen, and A. Metallinou, "Deep neural network acoustic models for spoken assessment applications," *Speech Communication*, vol. 73, pp. 14–27, 2015.

[3] Y. Qian, K. Evanini, X. Wang, C. M. Lee, and M. Mulholland, "Bidirectional lstm-rnn for improving automated assessment of non-native children's speech." in *INTERSPEECH*, 2017, pp. 1417–1421.

[4] T. M. Derwing, M. J. Munro, and M. Carbonaro, "Does popular speech recognition software work with esl speech?" *TESOL quarterly*, vol. 34, no. 3, pp. 592–603, 2000.

[5] H. Franco, V. Abrash, K. Precoda, H. Bratt, R. Rao, J. Butzberger, R. Rossier, and F. Cesari, "The sri eduspeaktm system: Recognition and pronunciation scoring for language learning," *Proceedings of InSTILL 2000*, pp. 123–128, 2000.

[6] A. Neri, C. Cucchiarini, and H. Strik, "Automatic speech recognition for second language learning: how and why it actually works," in *Proc. ICPhS*, 2003, pp. 1157–1160.

[7] Y. R. Oh and H. K. Kim, "A hybrid acoustic and pronunciation model adaptation approach for non-native speech recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 9, pp. 2379–2387, 2010.

[8] X. Wang and S. Yamamoto, "Second language speech recognition using multiple-pass decoding with lexicon represented by multiple reduced phoneme sets," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] F. Zhang and P. Yin, "A study of pronunciation problems of english learners in china," *Asian social science*, vol. 5, no. 6, pp. 141–146, 2009.

[10] F. Han, "Pronunciation problems of chinese learners of english." *ORTESOL Journal*, vol. 30, pp. 26–30, 2013.

[11] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, "Automatic detection of phone-level mispronunciation for language learning," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[12] 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 *International Workshop on Speech and Language Technology in Education*, 2009.

[13] W.-K. Lo, S. Zhang, and H. Meng, "Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[14] A. Lee and J. Glass, "A comparison-based approach to mispronunciation detection," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 382–387.

[15] W. Li, S. M. Siniscalchi, N. F. Chen, and C.-H. Lee, "Improving non-native mispronunciation detection and enriching diagnostic feedback with dnn-based speech attribute modeling," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6135–6139.

[16] 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 Communication*, vol. 51, no. 10, pp. 896–905, 2009.

[17] S. Mao, Z. Wu, R. Li, X. Li, H. Meng, and L. Cai, "Applying multitask learning to acoustic-phonemic model for mispronunciation detection and diagnosis in l2 english speech," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6254–6258.

[18] L. Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 25, 2015.

[19] J. Vaněk, J. Zelinka, D. Soutner, and J. Psutka, "A regularization post layer: An additional way how to make deep neural networks robust," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 204–214.

[20] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

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

[22] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," *INTERSPEECH*, 2018.

[23] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[24] V. Laborde, T. Pellegrini, L. Fontan, J. Mauclair, H. Sahraoui, and J. Farinas, "Pronunciation assessment of japanese learners of french with gop scores and phonetic information," 2016.