8th ISCA Workshop on Speech and Language Technology in Education (SLaTE 2019)
20-21 September 2019, Graz, Austria

Automatic Scoring Minimal-Pair Pronunciation Drills by Using Recognition Likelihood Scores and Phonological Features

Lei Chen[1]*, Qianyong Gao, Qiubing Liang, Jiahong Yuan*[1]*, Yang Liu*[1]

[1]LAIX Silicon Valley AI Lab, California, USA
LAIX Inc. Shanghai China

{lei.chen, qianyong.gao, qiubing.liang, jiahong.yuan, yang.liu}@liulishuo.com

Abstract

In the mispronunciation detection task using automatic speech recognition (ASR) technology, a recent trend is utilizing phonological features (PFs). PFs have an advantage in generating pronunciation correction feedback comparing with the likelihood features from the ASR framework. However, previous studies only compared PFs with one type of likelihood feature, goodness of pronunciation (GOP). In this paper, we conducted a more thorough comparison by including more likelihood features proposed in the previous literature. Our experiments showed that PFs brought additional performance gains over basic likelihood features, but not for the feature set containing log likelihood ratio (LLR) features. Our findings are helpful to the community for a better understanding of the contributions of PFs to the mispronunciation detection task.

1. Introduction

In language education, pronunciation training serves an important role. In the past two decades, computer-aided pronunciation training (CAPT) technology has been increasingly used to find pronunciation errors and provide corrective feedback on the spoken responses from second language (L2) learners [1]. Nowadays, given the widespread usage of mobile phones, CAPT Apps show their benefits for providing L2 learners ubiquitous access to pronunciation diagnosis.

Regarding detecting mispronunciations, a large number of research works and some commercial products have been developed on the principles of automatic speech recognition (ASR) using hidden Markov models (HMMs). For example, goodness-of-pronunciation (GOP) [2] is a commonly used method for measuring how well the uttered pronunciation matches with canonical transcription. The GOP score is a ratio between the expected phoneme's forced-alignment (FA) likelihood and the maximum phoneme recognition likelihood over the uttered speech segment. A higher GOP score means a higher probability that the expected phoneme was correctly pronounced.

Enhancements on top of the basic GOP scores have been proposed for more accurate and robust mispronunciation detection. For example, [3] proposed log-likelihood ratios (LLRs), which are the log-likelihood ratio computed using the model for any other phoneme and the model corresponding to the expected phoneme. In their mispronunciation detection experiments, LLRs were found to be superior to the basic GOP feature.

Different from the HMM likelihood features used in the GOP related methods, articulatory features (AFs) have received increasing attention. When pronouncing a phoneme, a speaker needs to change his or her vocal tract to approriate configurations. Clearly, these configurations, such as roundness of the lips, height of the tongue, and so on, can serve as useful traits for modeling the spoken phonemes. [4] is the first study that systematically utilized AFs to detect mispronunciations. In their research, the representation space consists of the following traits: *jaw*, *lip separation*, *lip rounding*, *tongue frontness*, *tongue height*, *tongue tip*, *velum*, and *voicing*. For each trait, an HMM was learned to estimate trait states from the audio signals directly. For example, lip separation consists of four states, including *closed*, *slightly apart*, *apart*, and *wide apart*. The authors found that adding AFs helps pronunciation evaluation. In addition to boosting mispronunciation detection performance, another advantage is that AF's closeness to real vocal track configurations can be used to provide useful corrective feedback conveniently. [5] extracted speech attributes (similar to the AF space introduced earlier) based on deep neural network (DNN) models from speech streams. Then, these speech attributes' posterior probability scores were used to form a decision tree (DT) for distinguishing correct and incorrect phoneme pronunciations. Paths in the DT can provide instructions to inform the L2 learners the reasons causing their wrong pronunciations. One way of modeling AF is via phonological features (PF) [6], which uses presence or absence of each articulatory trait to represent phonemes. Using such binary representations allows us to use a vector to concisely represent a phoneme's articulation configuration. Recently, [7] utilized PFs in mispronunciation detection and reported positive results. Compared to approaches used in [4, 5], only one DNN model was used.

In pronunciation training, one type of training is using minimal-pair drills. This is using two words with one contrastive feature, like /IH/ in bit and /IY/ in beat (short and long vowel /i/ sound) to train L2 learners to speak two contrasting sounds correctly. Compared to other drill types, the minimal-pair training can explicitly test one specific pronunciation skill and can be conducted in a short time. Therefore, such drill type has received increasing interest, especially in the mobile learning era [8].

In this paper, we will focus on evaluating mispronunciation detection on the minimal pair drills. In previous research, the PFs were compared with the GOP feature. However, it is not very clear whether such performance gain will retain when using more sophisticated likelihood features, e.g., LLRs described in [3]. Hence, we conducted the current study to fill this knowledge gap.

2. PF Model

The PF model was built based on the DNN acoustic model (AM) principles. Similar to a DNN AM, the PF model is a DNN using acoustic feature inputs of each audio frame. Instead of estimating the possibility of being an HMM tri-phone state, the PF model estimates possibilities of having phonolog-

10.21437/SLaTE.2019-6

PF	Meaning; is the segment	PF_{IH}
syl [syllabic]	the nucleus of a sylalble	1
son [sonorant]	produced with a relatively unobstructed vocal tract?	1
cons [consonantal]	consonantal (not a vowel or glide, or laryngeal consonant)?	0
cont [continuant]	produced with continuous oral airflow?	1
delrel [delayed release]	an affricate?	0
lat [lateral]	produced with a lateral constriction?	0
nasal [nasal]	produced with nasal airflow?	0
strident [strident]	produced with noisy friction?	0
voi [voice]	vocal folds vibrating?	1
sg [spread glottis]	vocal folds abducted?	0
cg [constricted glottis]	vocal folds adducted?	0
ant [anterior]	Is a constriction made in the front of the vocal tract?	0
cor [coronal]	Is the tip or blade of the tongue used to make a constriction?	0
distr [distributed]	is a coronal constriction distributed laterally?	0
lab [labial]	involved constrictions with or of the lips?	0
hi [high]	produced with the tongue body raised?	1
lo [low]	produced with the tongue body lowered?	0
back [back]	produced with the tongue body in a posterior position?	0
round [round]	produced with the lips rounded?	0
velaric [velaric]	produced with velaric airstream?	0
tense [tense]	produced with an advanced tongue root	1
long [long]	produced with extended duration	0

Table 1: 22 *phonological features provided by the PanPhon software; Table is based on [9]. The third column shows concrete PF vector for the phoneme /IH/. Note that "1" denotes the presence while "0" denotes the absence of a trait.*

ical feature traits. We used a newly released speech recognition deep-learning package, **PyTorch-Kaldi** [10], to develop the PF model by using the TIMIT speech corpus [11]. Note that here we intentionally used a native English corpus with standard phoneme pronunciations with an aim for obtaining accurate mappings from voices to PF vectors. If using learners' spoken words, the mapping relationships listed in Table 1 are hard to be applied. The PyTorch-Kaldi package provides an efficient solution to jointly use Kaldi [12] for its efficiency on ASR research and the flexibility of PyTorch for general-purpose deep learning research. For example, conventional DNN AMs are trained for solving a multi-class classification task – predicting the most likely tri-phone state. However, the PF model is trained for solving a multi-label prediction task – each PF trait serves as a label. Using the PyTorch-Kaldi toolkit can allow us to quickly change to such a task by modifying a few lines of Python codes in PyTorch.

PF vectors were generated by using the PanPhon package [9]. First, the TIMIT ASR recipe provided in the PyTorch-Kaldi toolkit was used to train a DNN-HMM system to forced-align our pronunciation data to obtain phoneme sequences. Then, the phoneme symbols in TIMIT ARPA format were converted to international phonetics association (IPA) format. At last, using PanPhon we generated PF features containing 22 traits based on IPA inputs. Table 1 lists all of the PF traits and their corresponding meaning.

As depicted in Figure 1, 13 dimension Mel-scaled filter bank log energies and their Δ and $\Delta\Delta$ values, with a context of +/- 5 frames form the input to the DNN model. On top of inputs, there are 4 hidden layers. Each hidden layer contains 1024 neurons and uses ReLU (rectified linear unit) non-linear activation. Also, batch-normalization [13] and dropout ($p = 0.5$) [14] were applied for more accurate predictions. The output layer contains 22 neurons with a sigmoid activation for estimating the probabilities of the presence of the PF traits. The DNN is trained to minimize the binary cross-entropy loss (BCE) between the PF probability outputs and the target values, which were constructed in PanPhone by marking the absence or presence of a phonological feature at a particular time frame with 0 or 1 respectively. A standard SGD optimization method was used to train the model with an initial learning rate (LR) of 0.08. The model was trained for 24 epochs and the learning rate was adjusted during the entire training process.

3. Experiment

3.1. Pronunciation Data

36 phonemes were considered in our study. For each phoneme, we defined its competing phonemes up to 3 based on our teaching principles. Table 2 lists several vowels and consonants with their competing phonemes.

Phoneme	Competing Phoneme(s)
AO	AA, AW, AH
IH	IY, EH
AW	AO
N	M, L, NG
P	B, F
T	D

Table 2: *Examples of the minimal pairs being considered in our study.*

We used two different data sets in our mispronunciation detection experiments. First we used a simulated data set that is generated from L1 reading data following [15]. From an audio book, we extracted spoken words read by a professional native speaker. Then, we generated mispronunciations by changing a spoken word's canonical transcription to a minimal pair version. For example, for word *bit*, we changed its canonical transcrip-

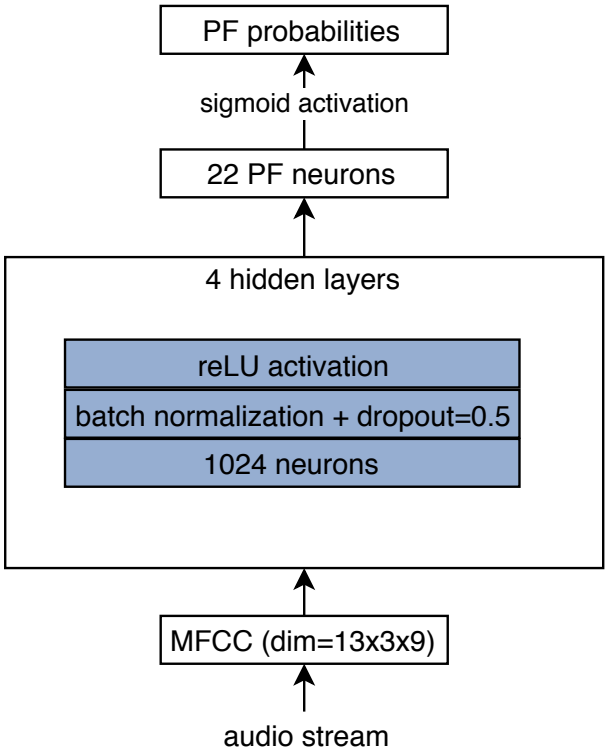

Figure 1: *PF DNN model*

Data	# Correct	# Incorrect
L1 data	266, 361	32, 937
L2 data w/ IPA	21, 937	3, 822

Table 3: *Information about the two pronunciation data sets*

tion to *beat* so that an artificial error of pronouncing long vowel /IH/ to short vowel /IY/ was created.

The second one is an L2 data set from our in-house large-sized L2 read-aloud samples. All these audio samples were from LAIX's Liulishuo English learning App. The majority of users are from China and they practice English reading on various types of mobile phones. On the sampled spoken sentences, human transcribers marked these sentence's IPA transcriptions. For the current study, we only focused on replacements with one phoneme difference based on IPA sequences. Table 3 summarizes these two pronunciation data sets, including their nature, the counts of phonemes with correct (positive) and incorrect (negative) pronunciations.

3.2. Implementations

As depicted in Figure 2, the minimal-pair pronunciation detection task is modeled as a classification task. The speech input and the corresponding transcriptions are used to run through a DNN-HMM recognizer to compute likelihood related features, which was built with the Kaldi open-source toolkit. This model is a 9-layer time delayed neural network (TDNN) using MFCC acoustic features from the current frame plus the previous and following 5 context frames. The ASR model was trained on our in-house read-aloud corpus containing about $2,500$ hours of native and non-native speech files. The ASR system achieved a word error rate (WER) of 9% on learners speech.

For a phoneme q_k, its corresponding audio observation is $\mathbf{O}$ with a duration of $d_{\mathbf{O}}$; its competing phonemes are denoted as q_i, where $i \in K$ and $i \neq k$. The GOP score feature [2] and generalized word posterior probability (GWPP) [16] are computed as follows.

$$GOP(\mathbf{O}, q_k) = \log\left(\frac{p(\mathbf{O}|q_k)}{\max_i(\mathbf{O}|q_i)}\right)/d_{\mathbf{O}} \quad (1)$$

$$GWPP(\mathbf{O}, q_k) = \log\left(\frac{p(\mathbf{O}|q_k)}{\sum_i(\mathbf{O}|q_i)}\right)/d_{\mathbf{O}} \quad (2)$$

In addition, the cost of running FA operation on $\mathbf{O}$ and the difference of duration of $\mathbf{O}$ to the standard norm values were included to provide 4 basic features.

For phoneme q_k, $LLR(\mathbf{O}, q_k, q_i)$ computes its log likelihood ratio (LLR) to another phoneme q_i. $LLRV(\mathbf{O}, q_k)$ is a vector containing all LLR values to other phonemes. When the phoneme q_k is a vowel, we used all other vowels to compute $LLRV$. Similarly, when the phoneme q_k is a consonant, we used all other consonants to compute $LLRV$.

$$LLR(\mathbf{O}, q_k, q_i) = \big(\log p(\mathbf{O}|q_k) - \log p(\mathbf{O}|q_i)\big)/d_{\mathbf{O}} \quad (3)$$

$$LLRV(\mathbf{O}, q_k) = [LLR(\mathbf{O}, q_k, q_i)], i \neq k \quad (4)$$

Using the PF DNN model trained on the TIMIT corpus, which was described in Section 2, we obtained a vector of PF probability vectors for each audio frame. Then, for each phoneme, the PF probabilities averaged over all of the included frames were used as additional features with a dimension of 22.

For each phoneme in our planned minimal-pair drill, several competing phonemes were pre-selected. Both correct pronunciations and mispronunciations were sampled from data sets for providing training and testing instances. Here, we used a gradient boosting tree model (GBM) classifier implemented in the scikit-learn Python package to classify the pronounced sound to one of the phonemes in the pool consisting of a phoneme and its competing phonemes. On both the simulated data set and the L2 data set, we run 10-fold cross validation (CV) experiment for each phoneme. Regarding evaluating minimal-pair pronunciation drill, for each phoneme, we measured false rejection rate (FRR) on correct pronunciations and false acceptation rate (FAR) on incorrect pronunciations. For a useful automatic scoring of minimal-pair pronunciation drill, these two values should be as small as possible. We then averaged all the phoneme's FRR and FAR values over the entire data set based on the percentage of phonemes to form our data-set level evaluation result.

3.3. Results

Table 4 reports the results on the simulated data set. Note that when estimating whether a pronunciation is correct or not, we simply used a threshold of 0.5 to convert the prediction posterior probabilities to decisions. Given the fact that much more positive cases (correct pronunciations) existed in the simulated data set, FRR values are very low.

Similar to the findings reported in [7], adding PFs helped to reduce both FRR and FAR of the baseline system using basic features (including a GOP score). However, adding LLRV provided larger FA reduction than adding PFs. For the enhanced system using both basic and LLRV features, adding PFs only can reduce FAR slightly.

On the L2 data, our initial experiment showed quite high FAR values more than 60.0%. When choosing the evaluation

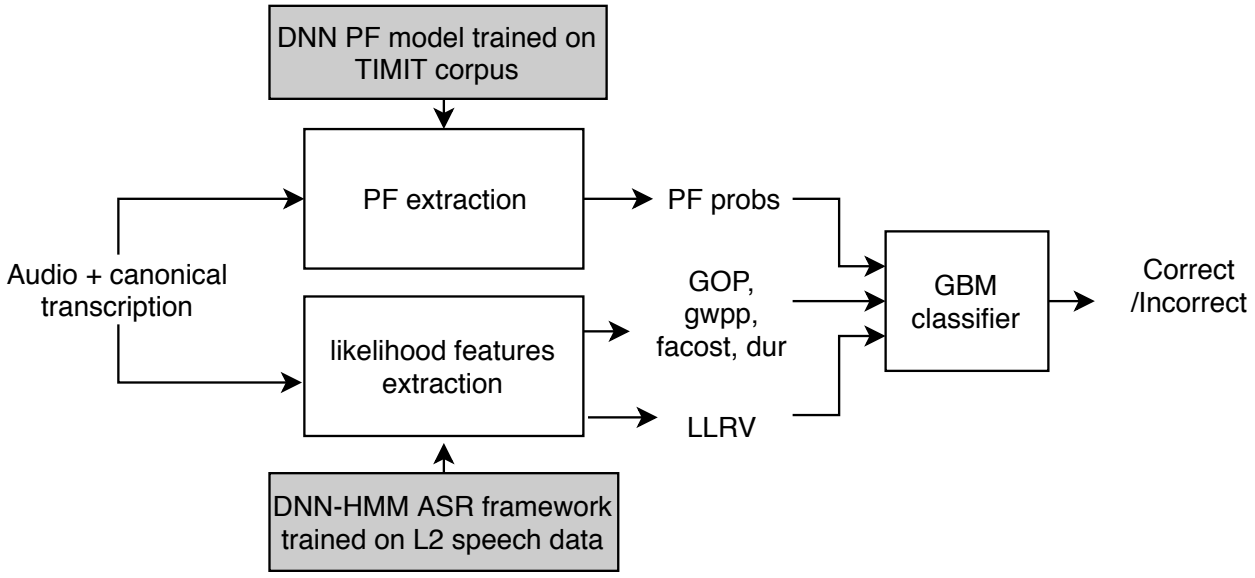

Figure 2: *Mispronunciation detection in the minimal-pair drill using both likelihood features and phonological features.*

Features	FRR	FAR
basic features	1.2%	40.5%
basic + LLRV	1.2%	29.0%
basic + PF	1.1%	36.8%
basic + LLRV + PF	1.2%	28.8%

Table 4: *Results on the simulated data set. FRR: false rejection rate on correct pronunciations; FAR: false acceptation rate on incorrect pronunciations.*

Features	FRR	FAR
basic features	12.9%	44.8%
basic + LLRV	11.0%	36.9%
basic + PF	13.3%	40.2%
basic + LLRV + PF	11.9%	36.6%

Table 5: *Results on the L2 speech data set.*

method used in a pronunciation training software, developers tend to control FRR in a small range while tolerating a high FAR. The reason of controlling FRR is to avoid rejecting learners' correct pronunciations and to reduce confusions or even frustrations brought to the learners. However, a very high FAR means that the evaluation method miss too many opportunities to correct learners' wrong pronunciations. Therefore, to tailor our method's behavior on the L2 data, we multiplied a penalty scaling factor on $p(Correct|\mathbf{O})$ to control FAR values. In the experiment, we applied a scaling factor of 0.3 and the obtained results are shown in Table 5. On real pronunciation data made by L2 learners, the patterns we observed in Table 4 still retain. Compared to PFs, LLRV showed larger FRR and FAR reductions over the baseline system using only the basic features. Since LLRV can be obtained by simple extension on the framework used for computing the basic features, incorporating them to achieving low FRR and FAR values has an advantage in terms of system complexity and development work. However, PFs have shown benefits on providing corrective feedback based on their tight connections to articulation. Utilizing them in mispronunciation detection and generating feedback is worth consideration.

4. Conclusions

On the minimal-pair pronunciation drills, we investigated using both likelihood features and phonological features to detect mispronunciations via a classification method. Consistent with previous findings, e.g., [7], PFs help to lower both FRR and FAR values when only using some basic features. However, LLRV features showed larger performance improvement than the PFs. Based on our finding, we will suggest including PFs in mispronunciation detection for their unique benefit on providing corrective feedback to help L2 learners.

We envision several future work directions. So far, the PF model only used a DNN and was trained on the TIMIT corpus. It is worth trying to train the model using a larger corpus and to utilize more advanced neural network models. In addition, PF features on each phoneme are just simply averaged from frame level estimations. More advanced feature integration method, e.g., [17] may worth trying.

5. References

[1] M. Eskenazi, "An overview of spoken language technology for education," *Speech Communication*, vol. 51, no. 10, pp. 832–844, 2009.

[2] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2, pp. 95–108, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639399000448

[3] S. Wei, G. Hu, Y. Hu, and R. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," *Speech Communication*, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639309000442

[4] J. Tepperman and S. Narayanan, "Using articulatory representations to detect segmental errors in nonnative pronunciation," *IEEE*

Transactions on Audio, Speech and Language Processing, vol. 16, no. 1, pp. 8–22, 2008.

[5] W. Li, K. Li, S. Siniscalchi, N. Chen, and C. Lee, "Detecting Mispronunciations of L2 Learners and Providing Corrective Feedback Using Knowledge-Guided and Data-Driven Decision Trees." in *Proc. of InterSpeech*, 2016. [Online]. Available: https://pdfs.semanticscholar.org/2953/c372757e510fcbb2fc64e69f2524d684076c.pdf

[6] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 333–353, 10 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230800901487

[7] V. Arora, A. Lahiri, and H. Reetz, "Phonological feature based mispronunciation detection and diagnosis using multi-task DNNs and active learning," in *Proc. of InterSpeech*, 2017. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:69032bc7-d9b0-45e6-b624-2822097a6f33

[8] B. Mak, J. Wong, J. Lo, M. Siu, M. Ng, Y.-C. Tam, Y.-C. Chan, K.-W. Chan, K.-Y. Leung, S. Ho, and F.-H. Chong, "PLASER: Pronunciation learning via automatic speech recognition," in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing* -, vol. 2. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 23–29. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1118894.1118898

[9] D. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. Levin, "Panphon: A resource for mapping IPA segments to articulatory feature vectors," in *Proc. of COLING*, Osaka, Japan, 2016, pp. 3475–3484. [Online]. Available: http://www.aclweb.org/anthology/C16-1328

[10] M. Ravanelli, T. Parcollet, and Y. Bengio, "The PyTorch-Kaldi Speech Recognition Toolkit," in *ICASSP*, 2019. [Online]. Available: http://arxiv.org/abs/1811.07453

[11] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167639390900107

[12] D. Povey, A. Ghoshal, G. Boulianne, and L. Burget, "The Kaldi speech recognition toolkit," in *Proc. IEEE Signal Processing Society*, 2011. [Online]. Available: https://infoscience.epfl.ch/record/192584

[13] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[14] N. Srivastava, G. Hinton, and A. Krizhevsky, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning*, 2014.

[15] S.-Y. Yoon, M. Hasegawa-Johnson, and R. Sproat, "Landmark-based automated pronunciation error detection," in *Proc. of InterSpeech*, 2010. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2010/i10_0614.html

[16] F. Soong and W. Lo, "Generalized word posterior probability (GWPP) for measuring reliability of recognized words," in *Proc. SWIM2004*, 2004. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Generalized+word+posterior+probability+(gwpp)+for+measuring+reliability+of+recognized+words#0%5Cnhttp://www.geocities.jp/wklojapan/doc/SoongSWIM2004.pdf

[17] W. Li, N. F. Chen, S. M. Siniscalchi, and C. Lee, "Improving Mispronunciation Detection for Non-Native Learners with Multisource Information and LSTM-Based Deep Models." in *Interspeech*, 2017. [Online]. Available: https://www.researchgate.net/profile/Marco_Siniscalchi/publication/319184805_Improving_Mispronunciation_Detection_for_Non-Native_Learners_with_Multisource_Information_and_LSTM-Based_Deep_Models/links/59ba2324458515bb9c48dc6f/Improving-Mispronunciation-Det