INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Explore Wav2vec 2.0 for Mispronunciation Detection

Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, Long Ma

Tencent Technology Co., Ltd, China
{xiaoshuoxu, yuetengkang, songjuncao, binghuailin, malonema}@tencent.com

Abstract

This paper presents an initial attempt to use self-supervised learning for Mispronunciaiton Detection. Unlike existing methods that use speech recognition corpus to train models, we exploit unlabeled data and utilize a self-supervised learning technique, Wav2vec 2.0, for pretraining. After the pretraining process, the training process only requires a little pronunciation-labeled data for finetuning. Formulating Mispronunciation Detection as a binary classification task, we add convolutional and pooling layers on the top of the pretrained model to detect mispronunciations of the given prompted texts within the alignment segmentations. The training process is simple and effective. Several experiments are conducted to validate the effectiveness of the pretrained method. Our approach outperforms existing methods on a public dataset L2-ARCTIC with a F1 value of 0.610.

Index Terms: Computer Assisted Pronunciation Training, Mispronunciation Detection, Self-supervised Learning

1. Introduction

Computer Assisted Pronunciation Training (CAPT) serves as a powerful tool for second-language (L2) learners to learn foreign languages. By recording and analyzing learners' speaking, it helps to provide specific feedbacks for L2 learners to improve their pronunciation. Mispronunciation Detection and Diagnosis (MDD), which pinpoints the erroneous pronunciation segmentation and provides phone-level diagnosis to users, is a key component of CAPT system. Owing to its potential applications, it has attracted lots of research interest over many years.

Goodness of Pronunciation (GOP) and its related likelihood-based features are extensively used in this field. Witt et al. extract GOP, normalized log posterior probability of given phones, to represent the quality of speaker's pronunciation [1]. With the advent of deep neural network (DNN), GOP is reformulated on DNN-based models, and Log Phone Posterior (LPP) is devised to evaluate the pronunciation [2, 3]. Recently GOP has been further enhanced by considering transition probabilities of HMM in [4]. These works mainly focus on Mispronunciation Detection (MD); hence no diagnosis is provided. Besides, by computing phone posterior probability approximately, text information is not explicitly used, while some search attempts to exploit text and phone information for modeling.

Li et al. combine speech attribute features with phone features to enhance the robustness and performance of MD [5]. Extended Recognition Network (ERN) [6], which uses a special decoding graph for phone recognition by utilizing phonological rules, could perform detection and provide detailed diagnoses. Nonetheless, ERN cannot cover all possible mispronunciations, as compiling too many rules leads to worse precision. Acoustic-Phonemeic Model (APM) and Acoustic Graphmic Model (APGM) resolve these limitations and perform free-phone recognition by utilizing phoneme and grapheme information [7, 8]. Compared to ERN, APM and APGM have considerable gains on CU-CHLOE, a home-built dataset for MDD. Further improvement has been achieved by using multi-task learning techniques in [9].

Recently, some researchers attempt to detect and diagnose mispronunciation errors directly through end-to-end pipelines. Leung et al. use CNN-RNN-CTC model for MDD [10]. Connectist Temporal Classification (CTC) [11] loss is applied for training, and Needleman-Wunsch Algorithm [12] is used to evaluate the performance. In [13], the authors integrate linguistic features into modeling, and the proposed model learns phonological rules implicitly from acoustic and linguistic features. Yan et al. extend phone set to accommodate non-categorical mispronunciation of L2 speaker [14]. [15] utilizes hybrid CTC-Attention models for acoustic modeling, and two decision methods are proposed for MD. These works train the model using CTC loss and require alignment algorithms to evaluate the performance in post-process.

The aforementioned methods usually use speech recognition corpus like TIMIT [16] and Librispeech [17] for acoustic modeling, and then the model is finetuned on limited L2 speech datasets. Since the final goal aims at detecting or diagnosing mispronunciations for L2 learners, non-native speech corpus is usually combined with native speech corpus for acoustic modeling. However, L2 learners pronounce words differently from L1 speakers [18]; thus it is difficult for people to transcribe their speech. Besides, current speech recognition requires thousands of transcribed audio to obtain good performance. Indeed, collecting and labeling such an amount of data requires lots of human efforts and costs.

Considering the difficulty and expense of collecting transcribed data for MD, we explore the possibility of using unlabeled audio for pretraining in this task. Specifically, for MD, we focus on the scenario of detecting mispronunciations of prompted texts. Motivated by the recent success of self-supervised learning method Wav2vec 2.0 on speech recognition [19], we utilize this technique to improve the performance of MD. We firstly train Wav2vec 2.0 model on unannotated data, and then linear classifiers are added on the top of the pretrained model for finetuning. Given the prompted text, roughly aligned segmentation and waveform, our model predicts whether the given text is correctly pronounced. That is, we formulate MD as a binary classification task in comparison to recent end-to-end methods that characterize MD as phone recognition [10, 15, 14]. We use the lately-published dataset L2-ARCTIC for experiments [20]. Experimental results show that even though unlabeled audio is used for pretraining, the model performs comparably to ASR pretraining methods, showing the advantages and effectiveness of self-supervised learning on MD. Our approach outperforms recent methods with a F1 value 0.610 on this dataset.

http://dx.doi.org/10.21437/Interspeech.2021-777

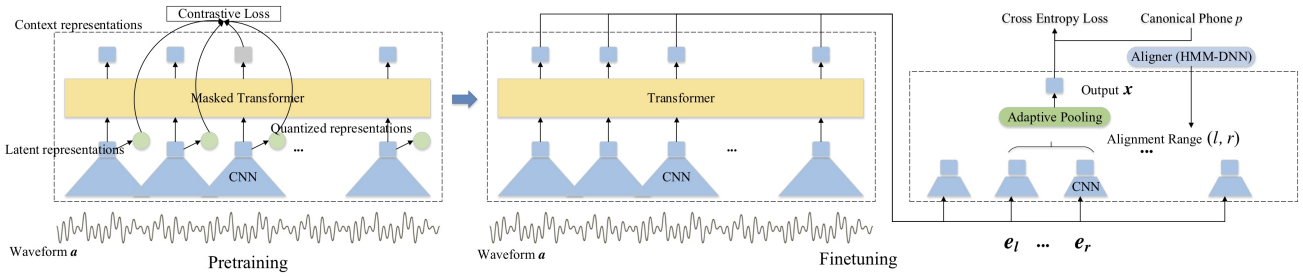

Figure 1: *Overview of pretrain and finetuning process. Quantized module is removed in the finetuning procedure, while new layers are introduced into the finetuning process.*

2. Proposed Method

We firstly use unannotated data to pretrain a Wav2vec 2.0 model. After the pretraining, the model is used to extract speech representations. A convolutional layer and an adaptive pooling layer are added on the top of the model to identify whether the prompted text is correctly pronounced. In the following section, we briefly review Wav2vec 2.0 [19], and introduce our approach.

2.1. Wav2vec 2.0 Pretraining

As shown in Figure 1, Wav2vec 2.0 mainly consists of a CNN encoder, a Transformer contextualized network and a quantization module. The input of model is raw wave, and the CNN encoder learns latent speech representations. The quantization module discretizes the latent representations to quantized representations. The training objective aims at recognizing the quantized representations using the output of contextualized network for each masked time step.

Please refer to [19] for more details of Wav2vec 2.0. The authors also report that the pretrained model helps to achieve state-of-the-art on phoneme recognition dataset TIMIT. As phoneme recognition is highly related to MD, we use this pretrained method in this task and expect performance gains by exploiting enormous unlabeled data.

2.2. Model Finetuning

Unlike pretraining, we have a little data for finetuning. For instance, L2-ARCTIC dataset only has about 3 hours of annotated audio. Hence, our finetuning principle lies in introducing new parameters into the pretrained model as few as possible. The network structure is illustrated in Figure 1. Given waveform a, the index of canonical phone p and roughly alignment range (l, r), we model the posterior probability of mispronunciations $P_\theta(t = 1|\mathbf{a}, p, l, r)$ parameterized by θ, where $t = 1$ represents mispronunciation, and $t = 0$ denotes correct pronunciation. Our modeling is different from the above-mentioned likelihood-based methods. (l, r) is explicitly included as a condition, and we model mispronunciations directly instead of computing phone posteriors.

During the finetuning, we remove the quantization module from the pretrained model, add a pointwise convolution and a max adaptive pooling on the top of Wav2vec 2.0 model. The pooling layer is applied to alignment time (l, r), which could be obtained by utilizing a trained HMM-DNN for force-alignment or annotated manually. Removing this extra process will be explored in our future work. Considering the output of the model is $\mathbf{x} \in R^N$ (N is the number of phones), we model the posterior probability $P_\theta(t = 1|\mathbf{a}, p, l, r)$ as below:

$$y = P_\theta(t = 1|\mathbf{a}, p, l, r) = \sigma(\mathbf{x}^p) \tag{1}$$

where σ is sigmoid function. Note that given the canonical phone (index by p), we use p^{th} dimension of $\mathbf{x}$ for posterior modeling. Furthermore, $\mathbf{x}$ is defined as:

$$\mathbf{x} = \max\{[\mathbf{W}\mathbf{e}_l + \mathbf{b}, \mathbf{W}\mathbf{e}_{l+1} + \mathbf{b}, \cdots \mathbf{W}\mathbf{e}_r + \mathbf{b}]\} \tag{2}$$

where $\mathbf{e}_l, \mathbf{e}_{l+1} \cdots \mathbf{e}_r$ are the outputs of Wav2vec 2.0, $\mathbf{W}, \mathbf{b}$ are parameters of the convolutional layer, and max operation is applied to time dimension.

The pointwise convolution combined with the max adaptive pooling could be viewed as N linear classifiers for N phones. Given the canonical phone, its classifier searches the alignment range (l, r) to get the highest value for probability modeling. The adaptive pooling drives the model to focus on pronunciation within the ranges; otherwise, the model becomes ambiguous on what to learn. With self-attention mechanism, the outputs have large receptive fields and capture both local and global dependencies, making them suitable representations for MD. Because when people perceive pronunciation errors, they do not listen to phones independently but feel acoustic changes within broad contexts. We also find that our model is not sensitive to alignment, while the accuracy of alignment often impacts the likelihood-based methods. In our primitive tests, the alignments ranges are shifted leftward or rightward by 50%, or the alignment durations are rescaled by a factor of $0.8, 0.9, 1.1$ or 1.2 respectively. Then we evaluate the model with the changed alignment ranges, but no significant change of performance is observed on the test set.

For training, cross-entropy loss is used to optimize the network:

$$L = -\{t \log y + (1 - t) \log (1 - y)\} \tag{3}$$

In inferencing, we readily use a threshold d for detecting, which is determinized optimally on the valid set.

$$\hat{t} = \begin{cases} 1 & y \geq d \\ 0 & y < d \end{cases} \tag{4}$$

$\hat{t}$ denotes our prediciton of mispronunciation.

3. Experimental Setting

3.1. Dataset

We use two public datasets Librispeech [17] and L2-ARCTIC [20] for experiments. The former is a widely-used native corpus

Table 1: *Details of dataset setup*

Corpus	Subsets	Spks.	Utters.	Hrs.
Librispeech	Train	5466	281231	960.98
L2-ARCTIC	Train	18	2699	2.78
	Valid	2	400	0.29
	Test	4	600	0.58

in speech recognition, and we mainly use this dataset for pretraining, while L2-ARCTIC is a non-native corpus built for MD. In our experiments, we split this dataset into a test set, a valid set and a training set. Four speakers (NJS, TLV, TNI, ZHAA) are used for testing, another two speakers (TXHC, YKWK) are chosen for validation, and the others are selected for training. The details of data division are shown in Table 1.

3.2. Experimental Details

We use Fairseq [21] and Kaldi [22] for experiments. The former is used to pretrain and finetune the model, while the latter is used to obtain the alignment and implement methods for comparison. The large architecture setting [19] is used to pretrain unlabeled data on Librispeech, which has 24 transformer blocks with encoding dimension 1024, linear dimension 4096 and 16 attention heads. During the finetuning process, we first fix parameters of the Wav2vec 2.0 model and only train the top layers (the components in the rightmost diagram of Figure 1) for 1000 iterations. Then this restriction is removed, and the model is finetuned jointly for another 7000 steps. Each iteration processes 1.28M samples, corresponding to 80s given a sampling rate 16KHz. The whole training process involves 8000 iterations, about 48 epochs. For each epoch, we enumerate a decision threshold from 0.1 to 0.9 and compute F1 values correspondingly, and the optimal threshold that achieves the highest F1 value is used for later inference. In testing, we feed required data into the model, which recognizes whether the given phone is correctly pronounced.

Kaldi is used to train a time-delay neural network (TDNN) from Librispeech corpus following the default setting[1], and then the network is finetuned on L2-ARCTIC training set. This model is used to generate alignment time for canonical phones and reimplement LPP [3] and Trans-GOP [4] for comparison. Their opensource codes are carefully adapted for MD on L2-ARCTIC dataset[2]. In detecting mispronunciations, we use a neural network based logistic regression referred as [3]. Logistic regression classifiers are trained based on each phone for Trans-GOP. Decision thresholds are then determined on L2-ARCTIC valid set.

3.3. Evaluation Metrics

For MDD, [8] provides a hierarchical structure with several metrics for evaluation. Since we merely focus on MD, parts of the metrics, Precision, Recall, F1, FAR and FRR are used for assessment. Precision is the fraction of correctly detected mispronunciations among the detected misprouncations, recall denotes the percentage of correctly detected mispronunciations, and F1 is the harmonic mean of precision and recall. These metrics are defined as below:

[1] `github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5`

[2] `github.com/sweekarsud`, `github.com/kaldi-asr/kaldi/tree/master/egs/gop_speechocean762`

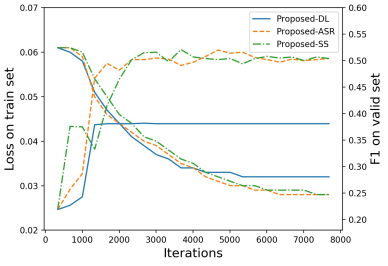

Figure 2: *Loss on train set and F1 value on valid set during the finetuning*

$$Pecision = \frac{TR}{TR + FR} \tag{5}$$

$$Recall = \frac{TR}{TR + FA} \tag{6}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

where TR represents the number of phones labeled as mispronunciations and detected as incorrect, FR is the number of phones annotated as correct pronunciation and identified as incorrect, and FA is the number of phones that are mispronounced but misclassified as correct.

Furthermore, FAR is the percentage of incorrect detection for mispronuciation (equal to one minus recall), while FRR is the percentage of incorret detection for correct pronuciation. They are defined as:

$$FRR = \frac{FR}{TA + FR} \tag{8}$$

$$FAR = \frac{FA}{FA + TR} = 1 - Recall \tag{9}$$

where TA is the number of phones labeled as correct pronunciation and detected as correct.

4. Result and Discussion

4.1. Effectiveness of Wav2vec 2.0 on MD

Firstly, we do experiments to validate the effectiveness of self-supervised learning on MD. We use three different schemes to train three models. Proposed-SS scheme utilizes Wav2vec 2.0 to pretrain the model on Librispeech and finetunes the model on L2-ARCTIC. Proposed-ASR scheme uses CTC loss to pretrain the model on Librispeech and finetunes on L2-ARCTIC. Proposed-DL scheme trains the model on L2-ARCTIC without any pretraining.

Figure 2 shows that Proposed-SS and Proposed-ASR schemes have a lower loss than that of Proposed-DL scheme after the coverage. Additionally, Proposed-DL, Proposed-ASR and Proposed-SS schemes achieve a F1 value of 0.381, 0.516 and 0.521 on valid set, respectively. Indeed these results show that though transcription is removed from pretraining, Wav2vec 2.0 pretraining improves performance for MD compared to training the model directly. Readers should also realize that there is a great discrepancy between Librispeech dataset and L2-ARCTIC dataset. The former mainly contains native speech recordings but the latter consists of non-native speech from different countries. Potential improvement could be achieved by

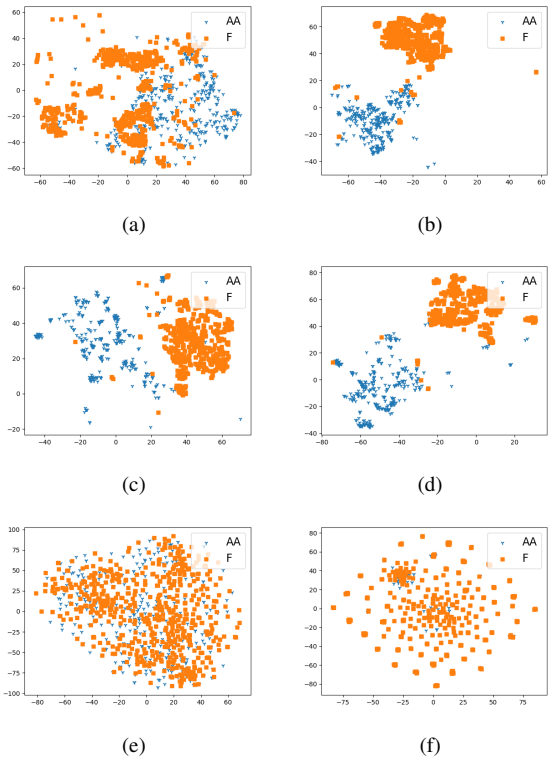

Figure 3: *Visulizations of the representations. (a), (c) and (e) refer to the visualizations of Proposed-SS, Proposed-ASR and Proposed-DL respectively before the finetuning, while (b), (d) and (f) are the visualizations of Proposed-SS, Proposed-ASR and Proposed-DL respectively after the finetuning.*

pretraining model on non-native audio, which could be gathered much easier than transcribed recordings.

4.2. Effects of Wav2vec 2.0 on Learned Representations

Viewing that self-supervised learning leads to comparable results like ASR pretraining, we explore how this technique contributes to this task. Since the three schemes use the same network structure but distinct initial network parameters, we visualize the representations of the penultimate layer, i.e. the outputs of Wav2vec 2.0 before and after the finetuning. t-SEN [23], a tools for displaying high-dimensional data, is used to visualize the representations. We randomly select two phones, AA and F, and extract corresponding representations within their alignment time for clear visualization (we choose those correctly pronounced cases only).

Figure 3 (b) and (d) show the representations of the two phones are grouped into two clusters after the finetuning for Proposed-SS and Proposed-ASR. Since convolution and adaptive pooling are applied to the representations within the alignment range, such operations could be viewed as linear classifiers for mispronunciation detection. For each canonical phone, correctly detecting mispronunciations means that its representations should be linear separable from that of other phones. Otherwise, the classifier could not detect substitution errors. In other words, our model works similarly to phone classification, though the training objective focuses on detecting pronunciation errors for each phone. It also explains the reason why ASR pretraining is quite helpful for our modeling, as speech recognition naturally learns to recognize (or classify) different phones.

Interestingly, as shown in Figure 3 (a) and (c), Wav2vec 2.0 pretraining and ASR pretraining perform similarly in doing representations, though in Wav2vec 2.0, the representations are not distinctly separated like those in ASR pretraining. Surprisingly, though only 2.78 hours of audio are used to finetune the model, the representations are grouped into two clusters (see Figure 3 (b)). By contrast, Figure 3 (e) and (f) reveal that without proper initializations, the representations of Proposed-DL distribute randomly before and after the finetuning. Overall, Wav2vec 2.0 pretraining learns discriminant speech representations, which help to improve performance for MD.

Table 2: *Performance on different methods on L2-ARCTIC test set.*

	Precision	Recall	F1	FAR	FRR
LPP	0.566	0.602	0.583	0.398	0.080
Trans-GOP	0.464	0.583	0.517	0.417	0.116
Proposed-ASR	0.538	**0.681**	0.602	**0.319**	0.101
Proposed-SS	**0.580**	0.643	**0.610**	0.357	**0.080**
Proposed-CTC	0.514	0.550	0.531	0.450	0.083

4.3. Comparison with Existing Methods

We reimplement recent methods LPP [3] and Trans-GOP [4] for comparison (see details of implementation in 3.2). Table 2 shows that our method outperforms these methods across all metrics and achieves a F1 value of 0.610 on L2-ARCTIC test set. Note that our proposed methods use unlabeled audio for pretraining, while the other methods use transcriptions for ASR pretraining. Another advantage of our approach lies in that we jointly train the detection model. However LPP and Trans-GOP rely on acoustic models for approximately computing phone posterior, and parameters of acoustic models are fixed when training detection models. Since our model uses more layers than LPP and Trans-GOP, we test deeper TDNN architectures in experiments but fail to find notable gains. These results demonstrate the effectiveness of Wav2vec 2.0 pretraining on MD.

An alternative for modeling mispronunciations could add a fully-connected layer on the top of the pretrained Wav2vec 2.0 model and use CTC loss for training as the recent works [10, 15, 14]. A model, denoted as Proposed-CTC, is trained in this way for comparison. During the training process, we evaluate the performance on the valid set, and the model that achieves the highest F1 score is used for testing. As shown in Table 2, this method works worse than Proposed-SS. We ascribe the degradation to two reasons. Firstly, Proposed-CTC does not exploit any prompted text for modeling. Besides, the outputs of this model are phone sequences; thus it is not easy to balance precision with recall to achieve a high F1 score.

5. Conclusions

In this paper, we explore Wav2vec 2.0 as pretraining for MD. With given canonical phones, alignment time and waveform, the model could be trained to recognize whether the pronunciation is correct or not. Experiments show that Wav2vec 2.0 pretraining learns discriminant features for phone classification, which provides good initializations for training. Our proposed method outperforms recent methods on L2-ARCTIC with a F1 value of 0.610, validating the effectiveness of Wav2vec 2.0 pretraining on MD.

6. References

[1] 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.

[2] W. Hu, Y. Qian, and F. K. Soong, "An improved dnn-based approach to mispronunciation detection and diagnosis of l2 learners' speech." in *SLaTE*, 2015, pp. 71–76.

[3] 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 Communication*, vol. 67, pp. 154–166, 2015.

[4] S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities." in *INTERSPEECH*, 2019, pp. 954–958.

[5] W. Li, N. F. Chen, S. M. Siniscalchi, and C.-H. Lee, "Improving mispronunciation detection for non-native learners with multisource information and lstm-based deep models." in *Interspeech*, 2017, pp. 2759–2763.

[6] 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.

[7] K. Li and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multi-distribution deep neural networks," in *The 9th International Symposium on Chinese Spoken Language Processing*, 2014, pp. 255–259.

[8] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 193–207, 2016.

[9] 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.

[10] W.-K. Leung, X. Liu, and H. Meng, "Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[12] V. Likic, "The needleman-wunsch algorithm for sequence alignment," *Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne*, pp. 1–46, 2008.

[13] Y. Feng, G. Fu, Q. Chen, and K. Chen, "Sed-mdd: Towards sentence dependent end-to-end mispronunciation detection and diagnosis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3492–3496.

[14] B.-C. Yan, M.-C. Wu, H.-T. Hung, and B. Chen, "An end-to-end mispronunciation detection system for l2 english speech leveraging novel anti-phone modeling," *arXiv preprint arXiv:2005.11950*, 2020.

[15] T.-H. Lo, S.-Y. Weng, H.-J. Chang, and B. Chen, "An effective end-to-end modeling approach for mispronunciation detection," *arXiv preprint arXiv:2005.08440*, 2020.

[16] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *STIN*, vol. 93, p. 27403, 1993.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[18] S. Cheng, Z. Liu, L. Li, Z. Tang, D. Wang, and T. F. Zheng, "Asr-free pronunciation assessment," *arXiv preprint arXiv:2005.11902*, 2020.

[19] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[20] G. Zhao, S. Sonsaat, A. O. Silpachai, I. Lucic, E. Chukharev-Khudilaynen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," *Perception Sensing Instrumentation Lab*, 2018.

[21] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[23] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.