AraS2P: Arabic Speech-to-Phonemes System

Bassam Mattar
Alexandria University
b.mattar@alexu.edu.eg

Mohamed Fayed
Applied innovation Center
m.essam@aic.gov.eg
Georgia Institute of Technology
mfayed8@gatech.edu

Ayman Khalafallah
Applied innovation Center
a.khalafallah@aic.gov.eg

Abstract

This paper describes AraS2P, our speech-to-phonemes system submitted to the Iqra'Eval 2025 Shared Task. We adapted Wav2Vec2-BERT via Two-Stage training strategy. In the first stage, task-adaptive continue pretraining was performed on large-scale Arabic speech-phonemes datasets, which were generated by converting the Arabic text using the MSA Phonetiser. In the second stage, the model was fine-tuned on the official shared task data, with additional augmentation from XTTS-v2-synthesized recitations featuring varied Ayat segments, speaker embeddings, and textual perturbations to simulate possible human errors. The system ranked first on the official leaderboard, demonstrating that phoneme-aware pretraining combined with targeted augmentation yields strong performance in phoneme-level mispronunciation detection.

1 Introduction

Automatic mispronunciation detection and diagnosis (MDD) plays a key role in computer-aided pronunciation learning (CAPL), providing learners with objective and scalable feedback on their pronunciation quality score (Kheir et al., 2023). In Arabic, MDD is particularly challenging due to the language's complex phonemic inventory, the presence of emphatic and pharyngeal consonants, and the semantic role of short vowels (diacritics) (Abdou and Rashwan, 2014). These characteristics make accurate phoneme-level detection especially important, as even subtle deviations can significantly change meaning.

In this work, we describe a system based on a Wav2Vec2-BERT architecture (Baevski et al., 2020) that employs a two-stage training strategy: (1) task-adaptive continue pretraining on large Arabic speech datasets—Common Voice (Arabic split), SADA, and MASC—using phoneme-level supervision generated via the MSA Phonetiser,[1] resulting in labeled corpora that capture fine-grained phonetic distinctions, and (2) fine-tuning on the official shared task data as well as targeted augmentation through XTTS-v2-synthesized recitations that vary in Ayat segments, speaker embeddings, and noisy textual content to simulate realistic recitations errors.

We summarize our contributions as follows:

- A phoneme-aware task-adaptive pretraining strategy for Arabic MDD using large-scale speech-phonemes data.
- A targeted augmentation pipeline where we add noise to text, convert the noisy text to phonemes using MSA-Phonetizer, and generate corresponding speech for many speakers using XTTS-v2 (Casanova et al., 2024).
- Our model ranks first on the Iqra'Eval 2025 benchmark leaderboard, demonstrating effectiveness of our training strategy.

2 Related Work

2.1 Arabic CAPL and Mispronunciation Detection

Computer-Assisted Pronunciation Learning (CAPL) systems rely on Mispronunciation Detection and Diagnosis (MDD) to provide automated feedback for learners (Witt and Young, 2000; Eskenazi, 2009). Early MDD approaches often derived pronunciation quality metrics from acoustic likelihoods computed from recognition results, such as the Goodness of Pronunciation (GOP) score (Witt and Young, 2000). While GOP provides a practical way to detect pronunciation deviations, its granularity is limited to the phone level and its accuracy can be affected by recognition errors. Other research (Bonaventura

[1] https://github.com/Iqra-Eval/MSA_phonetiser

Proceedings of The Third Arabic Natural Language Processing Conference, pages 469–474

et al., 2000; Raux and Kawahara, 2002) has enhanced pronunciation modeling by incorporating likely pronunciation variants into a pronunciation dictionary, which can involve manual specification of error patterns.

Recent years have seen the adoption of deep learning and end-to-end architectures for MDD, enabling systems to learn pronunciation error patterns directly from data (Peng et al., 2022). For Arabic, MDD poses additional challenges due to its rich consonant inventory, emphatic and pharyngeal sounds, and the omission of short vowels in most written text and ASR systems (Kheir et al., 2025). Consequently, slight pronunciation errors—such as mixing up emphatic and non-emphatic consonants—may change the meaning of a word.

Arabic MDD research has explored handcrafted acoustic features, CNN-based classifiers, and transfer learning from large-scale ASR models (Calık et al., 2023; Alrashoudi et al., 2025). Several works have focused on Qur'anic recitation, where precise phoneme articulation is central (Abdou and Rashwan, 2014; Alrumiah and Al-Shargabi, 2023; Harere and Jallad, 2023). (Kheir et al., 2025) provided the first publicly available benchmark for Arabic phoneme-level MDD, using Qur'anic recitation with time-aligned phoneme annotations.

2.2 Self-Supervised Phoneme Recognition Models

Self-supervised learning has significantly advanced phoneme recognition, which in turn has improved the performance of MDD systems. Wav2Vec-BERT 2.0 model (Baevski et al., 2020) learns contextualized speech representations from raw audio by combining a convolutional encoder with a Transformer context network (Devlin et al., 2019; Baevski et al., 2019). It was pretrained using a contrastive objective (Chen et al., 2020; He et al., 2020) over masked audio segments, then fine-tuned with a Connectionist Temporal Classification (CTC) objective (Graves et al., 2006). Wav2vec 2.0 achieves state-of-the-art performance in phoneme recognition tasks, making it well-suited for MDD.

Building on this, Wav2Vec-BERT integrates a BERT-style masked language modeling (MLM) objective (Devlin et al., 2019) with the Wav2Vec 2.0 framework (Chung et al., 2021). This joint optimization learns both quantized acoustic units and contextual relationships between them, producing richer and more discriminative phonetic representations. Instead of iteratively re-clustering discrete units like HuBERT (Hsu et al., 2021), w2v-BERT learns quantization and context modeling in a single end-to-end process.

Multilingual Wav2Vec-BERT 2.0 extends this approach to 143 languages using over 4.5 million hours of speech for pretraining (Barrault et al., 2023). Its large-scale multilingual exposure enables robust representation of fine phonetic distinctions, even in low-resource settings like Arabic MDD. Compared to Wav2Vec 2.0, Wav2Vec-BERT 2.0 incorporates MLM-based contextual modeling directly into the acoustic encoder, allowing it to learn longer-range phoneme patterns. For this reason, we used Wav2Vec-BERT 2.0 pretrained weights.

2.3 Benchmarks and Shared Tasks

Iqra'Eval Shared Task (Kheir et al., 2025) represents a milestone for Arabic MDD by offering a publicly available benchmark, standardized evaluation protocol, and a leaderboard for reproducible comparison. Similar to MGB Challenge for Arabic ASR (Ali et al., 2016) and other shared tasks in speech and Natural Language Processing (NLP), this benchmark has stimulated community engagement and methodological innovation. Through integrating controlled evaluation with phoneme-level detection, Iqra'Eval addresses a critical gap in Arabic CAPL research by establishing a standardized benchmark for systematic evaluation.

3 Two-Stage Training

We adapted Wav2Vec2-BERT (Barrault et al., 2023) to our downstream task via Two-Stage training. We continued pretraining it on Arabic speech-phonems pairs (section 3.1). Meanwhile, we conducted exploratory data analysis to measure the alignment between continue pretraining and fine-tuning (section 3.2). Finally, we utilized training set of the task as well as our synthetically generated dataset for fine-tuning (section 3.3).

3.1 Adaptive Continue Pretraining

Continue pretraining has shown to be an effective technique to improve the performance of pretrained models on languages of interest (Kalyan et al., 2021; Zhou et al., 2024; Fujii et al., 2024; Alves et al., 2024). To boost our model, we continued pretraining it on speech-phonemes pairs. We deployed MSA-phonetizer[2] to convert open-

[2] https://github.com/Iqra-Eval/MSA_phonetiser

source datasets with speech-text pairs into speech-phonemes pairs, hence adapting it to suite the downstream task (Adaptive Continue Pretraining). Specifically, our pretraining data is constructed from Common Voice Arabic split (Ardila et al., 2019), SADA(Alharbi et al., 2024) and MASC(Al-Fetyani et al., 2023) datasets. Table1 includes statistics about these datasets.

Dataset	size (hours)
Common Voice (Ar-Split)	157
SADA	668
MASC	1,000

Table 1: Statistics of datasets used in our adaptive continue pretraining stage.

We used Adam optimizer with weight decay (Loshchilov and Hutter, 2017). We set hyperparameters as follows: learning rate of 1×10^{-5}, Linear Decay scheduler, weight decay equals to 0.01, Adam betas of $(0.9, 0.999)$, gradient clipping at 1.0, and batch size of 32. We continue the pretraining for $800k$ iterations.

3.2 Exploratory Data Analysis

We have had a hypothesis that there is a discrepancy between pretraining data and fine-tuning one. So, we plotted the histogram of the most frequent phonemes in both the pretraining and training datasets. As shown in figure 1, the distributions of phonemes differ notably, particularly for elongated phonemes such as "aa," "ii," "uu," and "AA.". This observation confirms the correctness of our hypothesis and highlights the importance of further fine-tuning on downstream task.

Prior to fine-tuning, we notice a difference between the phoneme inventory in the training dataset and the phonemes produced by the MSA phonetizer. We align the phonemes as shown in Table 2.

3.3 Fine-tuning

After continuing pretraining, we performed vanilla fine-tuning for the model on our "Tuning dataset" 3.3.1. We used the same training parameters as that of continue pretraining.

3.3.1 Tuning dataset

To further align the model with the task, we used the training set provided with the task, and created synthetic dataset to increase overall data size. Preparing the synthetic data has went through two main stages: prepare the noisy text and generate corresponding audio files.

Phonetiser Phoneme	Inventory Phoneme
II0	II
I0I0	II
I0	I
I1	I
ii0	ii
i0i0	ii
i0	i
i1	i
UU0	UU
U0	U
U1	U
uu0	uu
u0u0	uu
uu1	uu
u0	u
u1	u

Table 2: Mapping from MSA phonetizer output to the training dataset phoneme inventory.

Prepare Noisy Text: We downloaded the text of the holy quran and perturbed the text with what we consider to be valid noise. The algorithm to generate valid noise is shown in algorithm1.

Algorithm 1 Noising Algorithm

```
procedure GENERATENOISYTEXT(text, arabic_chars, noise_map, max_noise)
    target_noise ← RandInt(1, max_noise)
    new ← empty list; count ← 0
    for ch in text do
        if count >= target_noise then
            Append ch
        else if UniformRandom(0, 1) < p_noise then
            count ← count + 1
            Choose noise type: delete / substitute / insert
            if substitute then
                Append RandChoice (noise_map[ch])
            else if insert then
                Append RandChoice(arabic_chars), ch
            end if
        else
            Append ch
        end if
    end for
    return Join(new)
end procedure
```

Audio Generation: We downloaded many audio files for various speakers to ensure the variety of data and to avoid overfiting over small set of speakers. Then, we generated speaker embeddings using embedder module in XTTS-v2 (Casanova et al.,

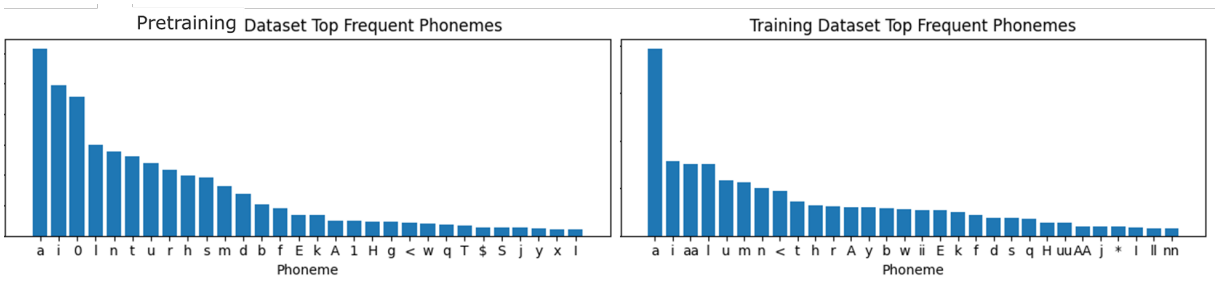

Figure 1: Histogram of top frequent phonemes in pseudo-labelled pretraining and training datasets

2024). Finally, we converted the noisy text to audio files using XTTS-v2.

The resulted dataset is 60 hours of audio files, and represented 30% of Tuning data.

While selecting checkpoint for testing, we noticed a shift in distribution between our valid set and competition's test set. Hence, we selected checkpoint saved after 2.5 epochs for submission to balance generalizability and good performance on the downstream task.

4 Results

In this section, we illustrate the metrics used (section 4.1), report quantitative results (section 4.2), and shows some examples from our qualitative analysis (section 4.3).

4.1 Metrics

The system is evaluated using several complementary metrics. First, the **Correct Rate** measures the proportion of phonemes that are detected correctly, and is defined as $1 -$ Phoneme Error Rate (PER). In addition, **Accuracy** captures the proportion of phonemes classified correctly as either pronounced correctly or mispronounced. To further distinguish system behavior, **True Acceptance (TA)** refers to cases where a correct phoneme is correctly accepted, while **True Rejection (TR)** corresponds to mispronounced phonemes that are correctly flagged. Conversely, errors are represented by **False Acceptance (FA)**, when a mispronunciation is missed, and **False Rejection (FR)**, when a correct phoneme is wrongly flagged. Beyond detection, **Correct Diagnosis (CD)** evaluates how often the system not only detects a mispronunciation but also identifies the specific mispronounced phoneme. Finally, the system's classification quality is summarized through **Precision**, defined as $\frac{TR}{TR+FR}$, **Recall**, defined as $\frac{TR}{TR+FA}$, and their harmonic mean, the **F1-score**, computed as $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.

4.2 Quantative Analysis

Table 3 shows the results of our system under different setups: after adaptive continue pretraining, fine-tuning on the official training data of the task, and after fine-tuning on our Tuning data. The results demonstrate that fine-tuning is essential for optimizing the system's alignment with Qur'anic recitation assessment. More importantly, they show the effectiveness of our synthetic data generation pipeline, achieving top performance across all of our systems.

4.3 Qualitative Analysis

Table 4 presents examples from both the fine-tuning on training set only setup and the continued pretraining-one. Because of time constraints and high similarity between fine-tuning on training set only and on Tuning set, we leave its qualitative analysis for future work. The results indicate that the system trained with pretraining alone fails to accurately predict phonemes associated with diacritics, particularly the "shadda". This limitation is likely due to the rarity of such phonemes in the pretraining data as discussed in subsection 3.1. This further confirms that adaptive continue pretraining was not sufficient and that we need for fine-tuning on the training set of the task.

5 Conclusion

In this work, we illustrated our recipe to adapt Wav2Vec-BERT 2.0 on speech-to-phoneme task. First, adaptively continued pretraining it on Arabic speech-phonemes corpora. Second, we prepared synthetic data for fine-tuning phase by generating noisy text, convert it to phonemes using MSA-phonetizer, and generate corresponding speech for many speakers using XTTS-v2. Our model scored first on IqraEval 2025, illustrating the ffectiveness of our approach.

System	F1↑	Prec.↑	Rec.↑	CR↑	Acc.↑	TA↑	FR↓	FA↓	CD↑
pretraining only	0.1923	0.1091	0.807	0.5156	0.5117	0.5264	0.4736	**0.193**	0.4363
fine-tuning									
training data	0.4561	0.3327	0.7252	0.8714	0.8576	0.8954	0.1046	0.2748	0.568
Tuning data	**0.4726**	0.3713	0.6501	**0.8985**	**0.8701**	**0.9209**	**0.0791**	0.3499	**0.6873**

Table 3: Performance on the Iqra'Eval 2025 leaderboard. CR = Correct Rate, Acc. = Accuracy, TA = True Acceptance, FR = False Rejection, FA = False Acceptance, CD = Correct Diagnosis.

Ref. Aya Segment	Recited Aya Segment (With Error)	Pretrained System Output	Fine-tuned System Output
يُغَشِّيكُمُ النُّعَاسَ أَمَنَةً مِنْهُ	يُخَشِّيكُمُ النُّعَاسُ أَمَنَةً مِنْهُ	ii x $ ii k m l n E aa s m n h m n h	y u x A $$ ii k u m u l nn u E aa s u < a m a n a t a n mm i n h u
إِيَّاكَ نَعْبُدُ	إِيَّاكَ نَعْبَدُ	y aa k n E b d	< ii y aa k a n a E b a d u
ذَلِكَ الْكِتَابُ لَا رَيْبَ فِيهِ	ذَلِكَ الْكِتَابُ لَا رَيْبُ فِيهِ	* aa l i k l k t aa b l aa r ii b f ii h	* aa l i k a l k i t a b u l aa r a y b a f ii h i
ٱلرَّحمَٰن	ٱلرَّرَّحمَٰن	a l r H m n	l rr rr rr a H m a n i

Table 4: Comparison Between Only Pretrained and Fine-tuned System

6 Acknowledgment

We thank the IqraEval organizers(El Kheir et al., 2025) for their support and for providing clear documentation and tools such as the evaluation API. We also acknowledge the contributors of the open-source datasets we used. This effort was supported by Applied Innovation Center (AIC) 's High Performance Computing facilities

References

Sherif Mahdy Abdou and Mohsen Rashwan. 2014. A computer aided pronunciation learning system for teaching the holy quran recitation rules. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 543–550. IEEE.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. Masc: Massive arabic speech corpus. In *IEEE-SLT*.

Sadeen Alharbi, Areeb Alowisheq, Zoltán Tüske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Raghad Aloraini, Raneem Alnajim, and 1 others. 2024. Sada: Saudi audio dataset for arabic. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10286–10290. IEEE.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. pages 279–284.

Norah Alrashoudi, Hend Al-Khalifa, and Yousef Alotaibi. 2025. Improving mispronunciation detection and diagnosis for non-native learners of the arabic language. *Discover Computing*, 28(1):1.

Sarah S Alrumiah and Amal A Al-Shargabi. 2023. Intelligent quran recitation recognition and verification: Research trends and open issues. *Arabian Journal for Science and Engineering*, 48(8):9859–9885.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, and 1 others. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar,

Justin Haaheim, and 1 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Patrizia Bonaventura, Daniel Herron, and Wolfgang Menzel. 2000. Phonetic rules for diagnosis of pronunciation errors. In *KONVENS 2000/Sprachkommunikation, Vorträge der gemeinsamen Veranstaltung 5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS), 6. ITG-Fachtagung" Sprachkommunikation"*, pages 225–230.

Sükrü Selim Calık, Ayhan Kucukmanisa, and Zeynep Hilal Kilimci. 2023. An ensemble-based framework for mispronunciation detection of arabic phonemes. *Applied Acoustics*, 212:109593.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and 1 others. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Youssef Elshahawy, Mostafa Shahin, and Ahmed Ali. 2025. Iqra'eval: A shared task on qur'anic pronunciation assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*. Association for Computational Linguistics.

Maxine Eskenazi. 2009. An overview of spoken language technology for education. *Speech communication*, 51(10):832–844.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *arXiv preprint arXiv:2404.17790*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Ahmad Al Harere and Khloud Al Jallad. 2023. Quran recitation recognition using end-to-end deep learning. *arXiv preprint arXiv:2305.07034*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023. Automatic pronunciation assessment–a review. *arXiv preprint arXiv:2310.13974*.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, and 1 others. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *arXiv preprint arXiv:2506.07722*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Linkai Peng, Yingming Gao, Binghuai Lin, Dengfeng Ke, Yanlu Xie, and Jinsong Zhang. 2022. Text-aware end-to-end mispronunciation detection and diagnosis. *arXiv preprint arXiv:2206.07289*.

Antoine Raux and Tatsuya Kawahara. 2002. Automatic intelligibility assessment and diagnosis of critical pronunciation errors for computer-assisted pronunciation learning. In *INTERSPEECH*, pages 737–740.

Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*.