Interspeech 2024
1-5 September 2024, Kos, Greece

Prompting Large Language Models with Mispronunciation Detection and Diagnosis Abilities

Minglin Wu[1,2], *Jing Xu*[1,2], *Xixin Wu*[1,2], *Helen Meng*[1,2]

[1]Department of System Engineering and Engineering Management,
The Chinese University of Hong Kong, HKSAR, China
[2]Centre for Perceptual and Interactive Intelligence Limited, HKSAR, China
{minglinwu, 1155170427}@link.cuhk.edu.hk, {wuxx, hmmeng}@se.cuhk.edu.hk

Abstract

Large Language Models (LLMs) have demonstrated significant achievements across diverse modalities. In this paper, we propose ATP-LLM, a framework that utilizes Audio and Text to Prompt LLMs to perform mispronunciation detection and diagnosis (MDD) tasks in second language (L2) English. ATP-LLM consists of an audio encoder and an LLM decoder. The audio encoder converts L2 English speech into speech representations digestible for LLMs. These speech representations, along with the corresponding canonical pronunciation, serve as audio and text prompts that enable the LLM decoder to generate the phones articulated by L2 English learners. Experiments show that our proposed ATP-LLM achieves a new state-of-the-art (SOTA) performance on the CU-CHLOE corpus with a Phone Error Rate (PER) of 8.56% and an F1 of 82.02%, outperforming the existing wav2vec2-CTC method whose PER and F1 are 8.98% and 80.93%, respectively.

Index Terms: L2 English, mispronunciation detection and diagnosis, prompting large language models

1. Introduction

Computer-Assisted Pronunciation Training (CAPT) offers people an economical and accessible way to learn and practice new languages. Central to CAPT is the mispronunciation detection and diagnosis (MDD) system, which is designed to identify incorrect pronunciations and provide targeted feedback. As depicted in Figure 1, a regular MDD pattern in second language (L2) English involves several steps. Initially, the CAPT system prompts learners to read a sentence. Then, the MDD system processes the spoken utterance to recognize the phones articulated. Finally, the MDD system aligns the recognized phone sequence and the canonical phone sequence to detect mispronunciations and give pinpointed feedback.

Recent studies in MDD have predominantly concentrated on two methodologies: (1) Pronunciation scoring, and (2) Phone recognition in L2 speech. In the former, researchers have utilized various confidence measures, such as Goodness of Pronunciation (GOP) [1] and its derivatives [2, 3], to evaluate pronunciations and detect mispronunciations with low scores. Some other researchers eschewed explicit pronunciation scoring in favor of end-to-end mispronunciation detection. For instance, Xu *et al.* [4] conceptualized MDD as a binary classification problem, while Zhang *et al.* [5] built a Transformer [6] based model to predict the error states of the phones. Nonetheless, pronunciation scoring is incapable of diagnosing the mispronunciations.

To overcome this limitation, the second category of research focuses on phone recognition in L2 speech. Extended recognition networks (ERNs) [7, 8] have been constructed to encompass both canonical pronunciations and potential mispronunciations, offering extra decoding paths for phone recognition. However, ERNs are inherently limited in their capacity to cover all mispronunciation variations. To mitigate this issue, some researchers try to integrate high-quality free phone recognizer into MDD systems. Leung *et al.* [9] designed a CNN-RNN-CTC architecture, combining a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN), and Connectionist Temporal Classification (CTC) loss [10], to facilitate phone recognition for MDD. Feng *et al.* [11] proposed a sentence-dependent MDD system that connects a sentence encoder and a sequence labeling model via an attention mechanism to incorporate the prompt sentence as extra textual information for better phone recognition. Furthermore, [12, 13] proposed to fine-tune the pretrained wav2vec 2.0 [14] with CTC loss (denoted as wav2vec2-CTC method in this paper) to perform phone recognition and MDD tasks in L2 English speech. The high correlation between phonemes and the learned audio representations in wav2vec 2.0 has propelled wav2vec2-CTC method to achieve state-of-the-art (SOTA) performance across multiple datasets.

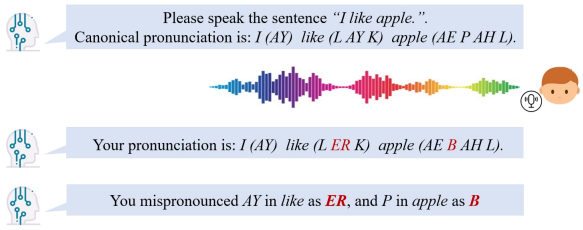

Figure 1: *A regular MDD pattern in L2 English. The pronunciations are represented using the CMU Pronunciation Dictionary.*

Recently, large language models (LLMs) have significantly enhanced performance across a range of natural language processing (NLP) tasks [15]. Despite these successes, the text-only interaction with artificial intelligence (AI) systems remains a limitation, falling short of the multimodal nature of human communication, which also encompasses sounds, images, and other sensory inputs. Therefore, plenty of researchers have been attracted to investigate the potential of LLMs in integrating various modalities, with the hope of getting overall performance enhancement. For instance, the Gemini model [16], as described by the Gemini team, is designed as a multimodal LLM and demonstrates enhanced performance across various modalities. Driess *et al.* developed PaLM-E [17], a system that integrates visual embeddings, neural 3D representations, and textual instructions within PaLM [18] to facilitate robotics tasks. Su *et al.* proposed PandaGPT [19] which combines the multi-modal

10.21437/Interspeech.2024-1772

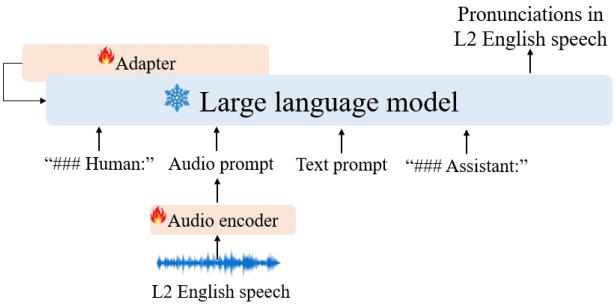

Figure 2: *Overall structure of the proposed ATP-LLM method.*

encoder ImageBind [20] and the Vicuna [21] to perform various tasks such as image description and question answering about audios. Huang *et al.* [22] built AudioGPT which takes ChatGPT as a coordinator to invoke different audio foundation models to perform different tasks. Fathullah *et al.* [23] extended the capabilities of LLMs to automatic speech recognition (ASR). By directly attaching an audio encoder to LLMs to digest the audio modality, they achieved competitive results compared to traditional ASR methods.

The success of above works and the interactive nature of language learning motivate us to explore the potential of LLMs in MDD. We propose ATP-LLM, a framework that utilizes speech representations and canonical pronunciations as Aduio and Text Prompts to make LLMs hear and recognize phones articulated by L2 English learners. With the alignment between recognized phone sequence and canonical phone sequence, we detect mispronunciations and give pinpointed feedback. Extensive experiments show the effectiveness of our proposed ATP-LLM framework. The contributions of this paper are threefold:

- This paper, to our best knowledge, is among the first ones that explore the potential of LLMs in addressing the challenging L2 English speech.
- This paper proposes ATP-LLM, a framework that utilizes audio and text prompts to prompt LLMs to perform MDD tasks and achieves a new SOTA performance on an L2 corpus.
- This paper investigates a variety of factors that influence the MDD abilities of LLMs.

The structure of the paper is as follows: Section 2 describes the proposed method. Section 3 presents the experiments and evaluations. Conclusion is given in Section 4.

2. Methods

Figure 2 shows the overall structure of the proposed ATP-LLM framework. ATP-LLM consists of an audio encoder, an LLM decoder, and an adapter. Specifically, the audio encoder converts raw L2 English speech into continuous speech representations digestible for LLMs. These speech representations, along with the canonical phone sequence corresponding to the sentence to pronounce, serve as audio and text prompts to make the LLM decoder recognize the actual pronounced phones. The recognized phone sequence is aligned with canonical phone sequence to achieve MDD. The adapter is responsible for efficiently adapting the parameters of the LLM decoder.

2.1. Audio encoder

As depicted in Figure 3, the audio encoder comprises a CNN feature extractor, a Transformer encoder, a prompt projection

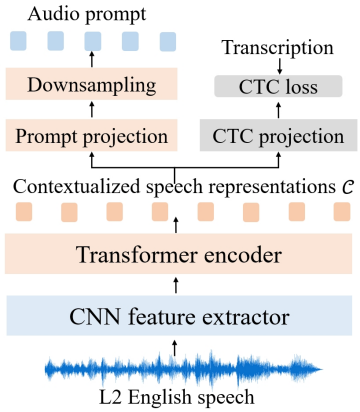

Figure 3: *The structure of the audio encoder.*

module, a downsampling module, and a CTC projection module. The CNN feature extractor consists of multiple convolutional layers that transform raw speech waveform into a sequence of feature vectors. Subsequently, the Transformer encoder encodes these feature vectors to produce contextualized speech representations, denoted as $\mathcal{C}$, which serve as inputs to two branches.

The first branch includes a prompt projection module followed by a downsampling module. The prompt projection module, comprising linear layers, transforms $\mathcal{C}$ into speech representations that are dimensionally compatible with LLMs. The downsampling module, comprising CNN layers, further down samples these representations to generate the final audio prompt. The second branch features a CTC projection module, which utilizes linear layers to project $\mathcal{C}$ into phone probabilities to compute an auxiliary CTC loss. The target of the CTC loss is the transcription of the L2 English speech. During inference, the second branch will be removed.

2.2. Large Language Model

In this paper, we adopt LLaMA 2 [24] as the LLM. LLaMA 2 is a versatile LLM with an auto-regressive Transformer architecture trained on trillions of tokens. As depicted in Figure 2, the inputs to LLaMA 2 include an audio prompt, a text prompt, and role information. The audio prompt is generated by converting raw L2 English speech using the audio encoder described in Section 2.1. The text prompt comprises the canonical phone sequences corresponding to the sentences intended for reading. Role information is used to simulate the interactive process, distinguishing the question part and the answer part. The output of LLaMA 2 is the pronunciations articulated in L2 English speech. We adopt Low-Rank Adaptation (LoRA) [25] as the adapter to adapt LLaMA 2.

2.3. Training strategy

We jointly train the audio encoder and adapt the LLM utilizing both the language modeling loss $\mathcal{L}_{LM}$, and the CTC loss $\mathcal{L}_{CTC}$. The overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{LM} + \lambda \mathcal{L}_{CTC} \tag{1}$$

where λ is a hyperparameter that controls the weight of the CTC loss. The language modeling loss $\mathcal{L}_{LM}$ aims to guide LLMs to correctly perform the next-token prediction task. We calculate

Table 1: *Details of CU-CHLOE*

	Training	Validation	Test	Total
Speakers	144	23	43	210
Hours	24	3.6	7	34.6

$\mathcal{L}_{LM}$ with Equation (2).

$$\mathcal{L}_{LM} = -\sum_{i=1}^{n} \log p(y_i|\boldsymbol{y}_{<i}; \boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{y}$ and $\boldsymbol{\theta}$ represent the transcribed phone sequences and the parameters of the entire model, respectively. The CTC loss $\mathcal{L}_{CTC}$ is calculated using Equation (3)

$$\mathcal{L}_{CTC} = -\log p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}_{AE}) \tag{3}$$

Here, $\boldsymbol{y}$ remains consistent with its definition in Equation (2), and $\boldsymbol{x}$ denotes the input L2 English speech. The term $\boldsymbol{\theta}_{AE}$ specifies the parameters of the CNN feature extractor, the Transformer encoder, and the CTC projection module in the audio encoder (see Figure 3).

During training, the parameters of the CNN feature extractor and LLaMA 2 are always fixed.

3. Experiments

3.1. Datasets

The corpus we utilize for the experiments is CU-CHLOE [26]. CU-CHLOE comprises 34.6 hours of L2 English speech from 100 Cantonese speakers (50 male, 50 female) and 110 Mandarin speakers (60 male, 50 female). Each speaker reads a set of 86 sentences, which are designed to cover all native English phonemes and can be categorized into four prompt types: (a) The AESOP's fable "The North Wind and the Sun", (b) Confusable words, (c) Minimal pairs, and (d) Phonemic sentences. Well-trained linguists annotated the corpus with phone-level transcriptions. The sampling rate of the speech is 16 kHz. Table 1 details the distribution of the corpus.

3.2. Experimental setups

The CNN feature extractor has 7 temporal convolutional layers that transform the raw speech signal into speech representations at a frequency of 49 Hz. These layers have 512 channels, strides of (5, 2, 2, 2, 2, 2, 2), and kernels of (10, 3, 3, 3, 3, 2, 2). The transformer encoder comprises 12 Transformer blocks, each with a model dimension of 768, a feed-forward dimension of 3072, and 8 attention heads. The weights of the CNN feature extractor and the Transformer encoder are initialized by fine-tuning the pre-trained wav2vec 2.0 on the dataset detailed in Section 3.1 using CTC loss, following the recipe detailed in [12].

The prompt projection module consists of one linear projection layer with an input dimension of 768 and an output dimension of 4096, matching the embedding dimension of LLaMA 2. The downsampling operation is optional. When it is applied, the downsampling module comprises one temporal convolutional layer with both input and output channels set at 4096. We utilizes kernels of (3, 10, 20) and corresponding strides of (2, 5, 10) to achieve varying downsampling rates. The CTC projection module has one linear projection layer with an input dimension of 768 and an output dimension of 32000, corresponding to the size of the LLaMA 2 tokenizer.

We adopt the LLaMA2-7B model for all experiments. With the rank set to 0, 16, and 32, the LoRA is applied to adapt the parameters of the query, key, value, and output projection matrices within the self-attention modules. Throughout the training process, the parameters of the CNN feature extractor are frozen, and those of the Transformer encoder are frozen for the initial 5000 updates. The hyperparameter λ in Equation (1) is set to 1.

We train the entire model using the Adam optimizer and the *WarmupDecayLR* learning rate schedule [27] with a peak learning rate of 1e-4 for 4 epochs on 2 A6000 GPUs with a batch size of 2.

3.3. Experimental results

3.3.1. Evaluation metrics

With the alignment between recognized phone sequence and transcribed phone sequence, we count the number of substitution (S), deletion (D), and insertion (I) errors. We then use Accuracy ((N-S-D-I)/N), which is 1-PER, and Correct Rate (CR, (N-S-D)/N) to evaluate the phone recognition in L2 English speech. N is the length of the transcribed phone sequence.

We adopt metrics derived from True Acceptance (TA), True Rejection (TR), False Rejection (FR), and False Acceptance (FA) [28] to evaluate the MDD performance. With the alignment among canonical, transcribed, and recognized phone sequences, TA occurs when both the human-transcribed phone and the recognized phone match the canonical pronunciation. TR occurs when both the transcribed phone and the recognized phone differ from the canonical pronunciation. FR occurs when the transcribed phone matches the canonical pronunciation, but the recognized phone does not. FA occurs when the recognized phone matches the canonical pronunciation, but the transcribed phone does not. We calculate False Rejection Rate (FRR, FR/(TA+FR)), False Acceptance Rate (FAR, FA/(FA+TR)), Detection Accuracy (DETA, (TA+TR)/(TA+FR+FA+TR)), Precision (TR/(TR+FR)), Recall (TR/(TR+FA)), and F1 score.

Within the TR cases, Correct Diagnosis (CD) means that the transcribed phone and the recognized phone are the same, while Diagnosis Error (DE) means that they are different. The derived Diagnosis Accuracy (DIAA) is CD/(CD+DE)

3.3.2. Main results

Table 2 shows the performance of different approaches. We take the performance of the wav2vec2-CTC method [12] as the baseline. The rank *R* determines the number of parameters that we can adapt using LoRA within LLaMA 2. Concretely, we have 0, 16.8, and 33.6 million of parameters to adapt with *R* being 0, 16, and 32, respectively. Notably, the proposed ATP-LLM model with an *R* = 32 configuration significantly outperforms the wav2vec2-CTC baseline across all evaluation metrics. For instance, our prompting method achieves an Accuracy of 91.44%, an FRR of 4.31%, an FAR of 18.73%, an F1 of 82.02%, and a DIAA of 90.03%, outperforming the wav2vec2-CTC method whose Accuracy, FRR, FAR, F1 and DIAA are 91.02%, 4.56%, 19.91%, 80.93%, and 89.35%, respectively.

Moreover, Table 2 shows that increasing *R* correlates with the performance improvement in the majority of the metrics. For example, the F1 increases from 81.59% at *R* = 0 to 81.87% at *R* = 16, and further to 82.02% at *R* = 32. Though the system performance in some metrics degrades with an increasing *R*, the difference is minimal. For instance, the FRRs for ATP-LLM at *R* = 16 and *R* = 32 are 4.29% and 4.31%, respectively, showing a mere 0.02% absolute difference.

Table 2: *Performance of different approaches. R is the rank in LoRA. Trainable parameters are the number of parameters that we can adapt using LoRA within LLaMA 2. R = 0 means that we are not using LoRA and the parameters of LLaMA 2 are totally frozen.*

Methods	Trainable parameters	Phone recognition (%)		Mispronunciation detection and diagnosis (%)						
		Accuracy↑	CR↑	FRR↓	FAR↓	Precision↑	Recall↑	F1↑	DETA↑	DIAA↑
wav2vec2-CTC	-	91.02	92.52	4.56	19.91	81.78	80.09	80.93	92.32	89.35
ATP-LLM R=0	0M	91.30	92.79	4.38	19.27	82.47	80.73	81.59	92.59	89.96
ATP-LLM R=16	16.8M	91.43	**92.96**	**4.29**	19.03	**82.80**	80.97	81.87	92.71	89.89
ATP-LLM R=32	33.6M	**91.44**	92.94	4.31	**18.73**	82.78	**81.27**	**82.02**	**92.75**	**90.03**

Table 3: *Investigating the impact of the text prompt. AP-LLM means that we only use the audio prompt and don't feed the canonical phone sequence as text prompt into the LLaMA 2.*

Methods	Trainable parameters	Phone recognition (%)		Mispronunciation detection and diagnosis (%)						
		Accuracy↑	CR↑	FRR↓	FAR↓	Precision↑	Recall↑	F1↑	DETA↑	DIAA↑
wav2vec2-CTC [12]	-	91.02	92.52	4.56	19.91	81.78	80.09	80.93	92.32	89.35
AP-LLM R=0	0M	91.22	92.74	4.55	19.37	81.87	80.63	81.25	92.44	89.65
AP-LLM R=16	16.8M	91.17	92.72	4.39	19.88	82.34	80.12	81.21	92.46	89.70
AP-LLM R=32	33.6M	91.25	**92.80**	4.43	19.43	82.28	80.57	81.42	92.52	89.92
ATP-LLM R=0	0M	**91.30**	92.79	**4.38**	**19.27**	**82.47**	**80.73**	**81.59**	**92.59**	**89.96**

Table 4: *The impact of downsampling on system performance. The number in parenthesis refers to the stride of the speech representations serving as audio prompt.*

Methods	Accuracy↑	FRR↓	FAR↓	F1↑	DETA↑	DIAA↑
wav2vec2-CTC [12]	91.02	4.56	19.91	80.93	92.32	89.35
ATP-LLM (200ms)	89.77	4.26	27.15	76.87	91.09	87.65
ATP-LLM (100ms)	91.24	4.05	20.74	81.32	92.59	90.18
ATP-LLM (40ms)	91.40	**4.01**	19.71	81.93	**92.80**	89.95
ATP-LLM (20ms)	**91.44**	4.31	**18.73**	**82.02**	92.75	**90.03**

Remarkably, even with $R = 0$, which means that we remove LoRA and totally freeze the parameters of LLaMA 2, our ATP-LLM model demonstrates superior performance to the wav2vec2-CTC baseline across all metrics. For instance, the Accuracy, F1 score, and DIAA are improved from 91.02%, 80.93%, and 89.35% to 91.30%, 81.59%, and 89.96%, respectively. This result suggests that our prompting method can effectively and efficiently introduce LLM to MDD even without compromising its original ability.

3.3.3. Ablation studies

We conduct extensive ablation studies to investigate the impact of the text prompt and downsampling on system performance.

As mentioned above, the text prompt in this paper refers to the canonical phone sequence corresponding to the sentence to pronounce. As depicted in Table 3, the text prompt significantly enhances the system performance. Notably, even with $R = 0$, ATP-LLM with text prompt outperforms all AP-LLMs without text prompts in nearly all metrics. For instance, the ATP-LLM ($R = 0$) achieves an Accuracy of 91.30%, an F1 of 81.59%, and a DIAA of 89.96%, marginally outperforming the AP-LLM ($R = 32$) recipe whose Accuray, F1, and DIAA are 91.25%, 81.42% and 89.92%, respectively. The only metric where AP-LLM ($R = 32$) excels is the CR, with a negligible absolute difference of 0.01%. These results show that ATP-LLM with a frozen LLaMA 2 generally outperforms AP-LLM with an adapted LLaMA 2. Furthermore, all AP-LLMs outperforms the wav2vec2-CTC baseline, demonstrating the LLM's capability to digest continuous speech representations and accurately model L2 English speech.

Table 4 shows the impact of downsampling speech representations that serve as audio prompt on system performance. Generally, ATP-LLM with a small audio prompt stride outperforms systems with larger strides. The original stride of the audio prompt output by the prompt projection module is 20ms. Increasing the stride from 20ms to 40ms results in a slight degradation in some metrics, while simultaneously yielding a modest improvement in other metrics. For instance, ATP-LLM (20ms) achieves an Accuracy of 91.44%, an F1 of 82.02%, and a DIAA of 90.03%, slightly outperforming the ATP-LLM (40ms) whose Accuracy, F1, and DIAA are 91.40%, 81.93% and 89.95%, respectively. While the FRR and DETA of ATP-LLM (20ms) are 4.31% and 92.75%, respectively, slightly inferior to the 4.01% FRR and 92.80% DETA for the ATP-LLM (40ms). Though increasing the stride to 200ms leads to a notable performance degradation, resulting in inferior performance to the baseline, the stride configuration at 100ms allows ATP-LLM to outperform the wav2vec2-CTC baseline across most metrics and remain competitive in others. The robustness of our ATP-LLM in relatively large audio prompt strides makes it possible to compress the audio sequence and benefits the LLMs on operating long audios.

4. Conclusions

In this paper, we propose ATP-LLM, a novel framework that uses audio and text to prompt large language models (LLMs) to perform mispronunciation detection and diagnosis (MDD) tasks. Extensive experiments show the effectiveness of ATP-LLM in modeling second language (L2) English. Notably, ATP-LLM significantly outperforms the existing wav2vec2-CTC method and achieves a new state-of-the-art (SOTA) performance. Further investigations reveal the influence of Low-Rank Adaptation (LoRA), text prompt, and downsampling on MDD ablities of ATP-LLM. Remarkably, ATP-LLM maintains superiority over the baseline across all metrics even with a totally frozen LLM, suggesting that we can invoke the MDD ability within LLMs without incurring additional costs. The inclusion of the text prompt significantly enhances system performance. While downsampling the audio prompt result in performance degradation, the system remains competent, thereby enabling the processing of long audios. In the future, we will enhance the distillation of knowledge within LLMs and further improve the system performance.

5. Acknowledgements

This work is supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd., a CUHK-led under the InnoHK scheme of Innovation and Technology Commission.

6. References

[1] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, pp. 95–108, 2000.

[2] H. Huang, H. Xu, Y. Hu, and G. Zhou, "A transfer learning approach to goodness of pronunciation based automatic mispronunciation detection," *The Journal of the Acoustical Society of America*, vol. 142, no. 5, pp. 3165–3177, 2017.

[3] 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.

[4] X. Xu, Y. Kang, S. Cao, B. Lin, and L. Ma, "Explore wav2vec 2.0 for Mispronunciation Detection," in *Proc. Interspeech 2021*, 2021, pp. 4428–4432.

[5] Z. Zhang, Y. Wang, and J. Yang, "End-to-end Mispronunciation Detection with Simulated Error Distance," in *Proc. Interspeech 2022*, 2022, pp. 4327–4331.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[7] X. Qian, H. M. Meng, and F. K. Soong, "The use of dbn-hmms for mispronunciation detection and diagnosis in l2 english to support computer-aided pronunciation training," in *INTERSPEECH*, 2012.

[8] Y.-B. Wang and L.-s. Lee, "Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 564–579, 2015.

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

[10] 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*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

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

[12] M. Wu, K. Li, W.-K. Leung, and H. Meng, "Transformer Based End-to-End Mispronunciation Detection and Diagnosis," in *Proc. Interspeech 2021*, 2021, pp. 3954–3958.

[13] L. Peng, K. Fu, B. Lin, D. Ke, and J. Zhan, "A Study on Fine-Tuning wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis," in *Proc. Interspeech 2021*, 2021, pp. 4448–4452.

[14] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[15] OpenAI, "Gpt-4 technical report," 2024.

[16] G. Team, "Gemini: A family of highly capable multimodal models," 2023.

[17] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," 2023.

[18] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, and et al, "Palm: Scaling language modeling with pathways," 2022.

[19] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "Pandagpt: One model to instruction-follow them all," 2023.

[20] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," 2023.

[21] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[22] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu, Y. Ren, Z. Zhao, and S. Watanabe, "Audiogpt: Understanding and generating speech, music, sound, and talking head," 2023.

[23] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, "Prompting large language models with speech recognition abilities," 2023.

[24] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, and et al., "Llama 2: Open foundation and fine-tuned chat models," 2023.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[26] H. M. Meng, W. K. Lo, A. M. Harrison, P. Lee, K. H. WONG, W.-K. Leung, and F. Meng, "Development of automatic speech recognition and synthesis technologies to support chinese learners of english : The cuhk experience development of automatic speech recognition and synthesis technologies to support chinese learners of english : The cuhk experience," 2010.

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, and et al, "Huggingface's transformers: State-of-the-art natural language processing," 2020.

[28] X. Qian, H. Meng, and F. Soong, "Capturing l2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (capt)," in *2010 7th International Symposium on Chinese Spoken Language Processing*, 2010, pp. 84–88.